

**EVALUACION DE ALGORITMOS DE SUSTRACCION DE  
FONDO PARA CONTEO DE PERSONAS**

**LUIS FERNANDO ESCOBAR TAFURT**

**UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS  
SANTIAGO DE CALI  
2016**

# **EVALUACION DE ALGORITMOS DE SUSTRACCIÓN DE FONDO PARA CONTEO DE PERSONAS**

**LUIS FERNANDO ESCOBAR TAFURT**

**Trabajo de grado presentado como requisito para optar al título de  
Ingeniero de Sistemas**

**Directores**

**Ing. MARIA PATRICIA TRUJILLO URIBE, Ph.D.  
Ing. DANIEL BARRAGÁN CALDERÓN, M.Sc.  
ÁREA DE PROCESAMIENTO DE IMAGENES**

**UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS  
SANTIAGO DE CALI  
2016**

**Nota de aceptación:**

---

---

---

---

---

---

\_\_\_\_\_ – Jurado

---

\_\_\_\_\_ – Jurado

---

**Prof. Maria Patricia Trujillo Uribe. Ph.D. –  
Director**

---

**Prof. Daniel Barragán Calderón. M.Sc. –  
Codirector**

**Santiago de Cali, 2016**

## AGRADECIMIENTOS

*Quiero expresar mis más profundos y sinceros agradecimientos a mi familia por la formación y cariño recibido durante todos estos años, por su apoyo y respaldo incondicional.*

*A Maria Trujillo y Daniel Barragán, directores de esta tesis por sus valiosas orientaciones en el transcurso de esta investigación, pero sobre todo por su confianza y apoyo.*

*Por último a todas aquellas personas que de una u otra forma nos sirvieron de inspiración y nos acompañaron durante este largo proceso...*

# CONTENIDO

	Pág.
<b>CAPÍTULO 1. <i>INTRODUCCIÓN</i></b>	<b>12</b>
<b>1.1 OBJETIVOS</b>	<b>13</b>
1.1.1 Objetivo General	13
1.1.2 Objetivos Específicos	13
1.1.3 Objetivos Estratégicos	13
<b>1.2 PROBLEMÁTICA</b>	<b>13</b>
<b>1.3 TRABAJO DESARROLLADO</b>	<b>14</b>
<b>1.4 BREVE INDICACIÓN DEL CONTENIDO DE CADA CAPÍTULO</b>	<b>15</b>
<b>CAPÍTULO 2. <i>MARCO CONCEPTUAL</i></b>	<b>16</b>
<b>2.1 INTRODUCCIÓN</b>	<b>16</b>
<b>2.2 ANTECEDENTES</b>	<b>16</b>
2.2.1 Antecedentes Locales	16
2.2.2 Antecedentes Nacionales	16
2.2.3 Antecedentes Internacionales	17
<b>2.3 BACKGROUND SUBTRACTION</b>	<b>17</b>
2.3.1 Modelado de fondo	19
2.3.2 Inicialización de fondo	19
2.3.3 Mantenimiento de fondo	19
2.3.4 Detección de primer plano	21
<b>2.6 CONCLUSIONES</b>	<b>23</b>
<b>CAPÍTULO 3. <i>REVISIÓN DE LITERATURA</i></b>	<b>24</b>
<b>3.1 INTRODUCCIÓN</b>	<b>24</b>
<b>3.2 MODELADO DE FONDO TRADICIONAL</b>	<b>24</b>
<b>3.3 MODELADO DE FONDO RECIENTE</b>	<b>26</b>
<b>3.4 IMPLEMENTACION EN TIEMPO REAL</b>	<b>27</b>
<b>3.5 CONCLUSIONES</b>	<b>28</b>
<b>CAPÍTULO 4. <i>ELECCIÓN DE ALGORITMOS</i></b>	<b>29</b>
<b>4.1 INTRODUCCIÓN</b>	<b>29</b>
<b>4.3 STATIC FRAME DIFFERENCE</b>	<b>29</b>
<b>4.4 KERNEL DENSITY ESTIMATION</b>	<b>30</b>
<b>4.5 MOG</b>	<b>31</b>
<b>4.6 CONCLUSIONES</b>	<b>31</b>

<b><i>CAPÍTULO 5.   PARAMETROS Y CRITERIOS DE COMPARACIÓN</i></b>	<b>33</b>
5.1     INTRODUCCIÓN	33
5.2     CONJUNTO DE VIDEOS DISPONIBLES	33
5.3     CRITERIOS DE COMPARACIÓN	35
5.4     PARAMETROS DE LOS ALGORITMOS	36
<b><i>CAPÍTULO 6.   PRUEBAS Y RESULTADOS</i></b>	<b>37</b>
6.1     INTRODUCCIÓN	37
6.2     PREPROCESAMIENTO DE VIDEOS	37
6.2.1     Ajuste de resolución	38
6.2.2     Aplicación de filtro	38
6.3     PRUEBAS DE MEMORIA VS TIEMPO	38
6.4     PRUEBAS DE PRECISIÓN Y RECALL	44
<b><i>CAPÍTULO 7.   CONCLUSIONES</i></b>	<b>50</b>
REFERENCIAS BIBLIOGRÁFICAS	51

## LISTA DE TABLAS

	Pág.
Tabla 6-1 Precisión y recall sobre los primeros 5 clips (promedio)	44
Tabla 6-2. Precisión y recall del 6 clip	45
Tabla 6-3 Precisión y recall de clips 7, 8 y 9	47
Tabla 6-4 Precisión y recall de clip 10	49

## LISTA DE FIGURAS

	Pág.
Figura 1-1 Diagrama de bloques del sistema desarrollado.	15
Figura 2-1 Etapas generales de sustracción de fondo.	18
Figura 2-2 Proceso de sustracción de fondo N es el número de frames que son usados para iniciar el fondo Bt y It son el fondo y la imagen actual en el tiempo t respectivamente.	22
Figura 4-1. Flujo de un algoritmo de diferencia de frames	30
Figura 5-1. Conjunto de videos utilizados en las pruebas	33
Figura 5-2. Videos nocturnos y de baja iluminación.	34
Figura 5-3. Videos con alta densidad poblacional.	34
Figura 5-4. Videos sin cámara estática	35
Figura 5-5 Video con Ondeo de arboles	35
Figura 6-1. Diagrama de flujo Pre procesamiento del video	37
Figura 6-2. Proceso de modificación de resolución	38
Figura 6-3 Memoria vs tiempo clip 1	39
Figura 6-4 Memoria vs tiempo clip 2	39
Figura 6-5 Memoria vs tiempo clip 3	40
Figura 6-6 Memoria vs tiempo clip 4	40
Figura 6-7 Memoria vs tiempo clip 5	41
Figura 6-8 Memoria vs tiempo clip 6	41
Figura 6-9 Memoria vs tiempo clip 7	42
Figura 6-10 Memoria vs tiempo clip 8	42
Figura 6-11 Memoria vs tiempo clip 9	43
Figura 6-12 Memoria vs tiempo clip 10	43
Figura 6-13 MOG sobre el clip 10	44
Figura 6-14 Algoritmo Frame difference en el clip 6	45
Figura 6-15 Algoritmo KNN en el clip 6	46
Figura 6-16 Algoritmo MOG2 en el clip 6	46
Figura 6-17 Algoritmo KNN en el clip 8	46
Figura 6-18 Algoritmo MOG2 en el clip 9	47
Figura 6-19 Algoritmo Frame Difference en el clip 7	47
Figura 6-20 Algoritmo KNN en el clip 10	48
Figura 6-21 Algoritmo Frame Difference en el clip 10	48



## LISTA DE ECUACIONES

	Pág.
Ecuación 1-1	19
Ecuación 1-2	20
Ecuación 1-3	20
Ecuación 1-4	20
Ecuación 4-1	29
Ecuación 4-2	30
Ecuación 4-3	30
Ecuación 4-4	31
Ecuación 4-5	31
Ecuación 4-6	32
Ecuación 3-7	32
Ecuación 3-8	32

## RESUMEN

La sustracción de fondo es una herramienta útil en el campo de la manipulación y análisis de videos, esta nos ofrece una técnica para separar los objetos en movimiento de los objetos estáticos, es ampliamente usado por herramientas como el Kinect, el conteo de personas, el tracking, entre otras.

En este documento se consigna la implementación de tres algoritmos, utilizados en sustracción de fondo para conteo de personas, y una posterior comparación bajo unos parámetros, se utilizara el lenguaje Python con la librería opencv, la lectura del video puede realizarse por medio de archivos o directamente desde la cámara web, a continuación se transforma a escala de grises para facilitar su procesamiento y manipulación, por último, por medio de las técnicas que se introducirán en capítulos posteriores, se aplica la sustracción de fondo y los diferentes tipos de pruebas para determinar las mejores condiciones de trabajo y según unos criterios definidos, comparar los algoritmos entre sí.

Las condiciones de operación consisten en tener una cámara estática, una iluminación de poca variabilidad, una resolución mínima de 950x540 píxeles, en los clips utilizados se evitan objetos que puedan influir en el resultado como autos en movimiento, motos, lluvia entre otros.

Los algoritmos fueron probados con 10 videos en diferentes condiciones bajo la misma resolución, obteniendo así resultados para la evaluación de desempeño y concluyendo cuál de los 3 se comporta mejor bajo que parámetros.

En general el trabajo desarrollado reúne diversas técnicas de procesamiento de video, sustracción de fondo, así como algoritmos propios orientados a un fin específico, pero que pueden ser utilizados para el desarrollo de otros proyectos afines o que incluyan el procesamiento de video.

**Palabras claves.** Sustracción de fondo, Python, OpenCv.

## ABSTRACT

Background subtraction is a very useful tool in video analysis and processing, this give us the way to split the moving objects from the static ones, it is widely used by tools like the Kinect, people counting, tracking, among others.

This document registers the implementation of three algorithms, used on background subtraction for counting people, followed by a comparison under some parameters; python will be used with the OpenCv library. The video reading can be done by files reading or directly from a web cam. Furthermore, it is converted to grayscale to make it easier to process and handle, finally, by different techniques that will be review on further chapters, background subtraction is applied and different test to figure out the best working conditions and by some defined criteria, compare each other.

The operation condition for the videos, consist on having a static camera, a minimum illumination change, a minimum resolution of 960x540 pixels, objects that can influence the result are avoided like moving cars, motorcycle, rain among others.

Each algorithm was tested with 10 videos on different conditions under the same resolution, getting results for the performance evaluation and concluding which of the three behaves in a better way under some parameters.

In general the developed work is a compendium of diverse techniques of video processing, background subtraction, and own algorithms all of them applied for a specific task, but it can be used for the development of other compatible projects or those that include the video processing.

**Keywords.** Background subtraction, Python, Opencv.

## Capítulo 1

---

# INTRODUCCIÓN

Cuando hablamos de procesamiento de video es necesario aplicar diferentes técnicas separadas en etapas para separar los píxeles que contienen información (objetos o personas en movimiento) de los objetos estáticos (conocido como fondo), entre estas encontramos la sustracción de fondo, que es una técnica ampliamente utilizada en el procesamiento de video para la segmentación (separar el primer plano o frente del fondo), obteniendo como resultado, píxeles ricos en información en los cuales se pueden aplicar diferentes técnicas como el conteo de personas o el tracking.

Por la naturaleza de esta técnica, ha sido ampliamente adoptada en áreas como la detección de movimiento, aplicaciones multimedia, vigilancia por medio de videos. En todos estos casos es necesaria la detección de objetos en movimiento en una escena.

Aunque la detección de movimiento por medio de sustracción de fondo no es siempre tan fácil como parece. Debido a algunos inconvenientes que surgen en los videos como son la señal pobre, ruido causado por la baja resolución de las cámaras, técnicas de compresión o un ambiente ruidoso, se pueden presentar numerosos falsos positivos. Los falsos positivos también pueden llegar producto de cambios de iluminación (gradual o repentina), un fondo dinámico (ondas del agua, arboles sacudidos por el viento), o jitter de la cámara por nombrar unas cuantas. Por otro lado, pueden ocurrir también falsos negativos cuando un objeto en movimiento esta hecho de colores similares a los del background (el también llamado efecto camuflaje). Con tales escenarios, una simple diferencia de frames con threshold global revela una solución poco robusta. Con el fin de afrontar estos desafíos, aparecen en la última década numerosos modelos de fondo y medidas de distancia para esquemas de optimización. Esos métodos son (al menos en teoría) más robustos frente al ruido e inestabilidad de fondo que los enfoques básicos de sustracción de fondo. Pero son reales? Y si lo son, que tan buenos son? Son aplicables en aplicaciones de tiempo real? Pueden ser implementados en arquitecturas ligeras?

En este proyecto, comparamos 3 de los algoritmos de sustracción de fondo que más se aplican en el conteo de personas bajo varios videos reales representando diferentes desafíos presentes en situaciones comunes. Las metas de este estudio son las siguientes:

- 1 Evaluar que tan bien se desempeñan algoritmos clásicos frente a algoritmos más sofisticados.
- 2 Comparar el poder de procesamiento y la cantidad de memoria requerida por cada método en tiempo de ejecución
- 3 Determinar qué tipo de situación o video se adapta mejor a cada método

En términos muy generales el procedimiento consiste de la captura de videos en diferentes situaciones, seguido del respectivo procesamiento de los datos, y la aplicación de los algoritmos de sustracción de fondo mostrando las zonas donde ocurre movimiento. Estos algoritmos pueden ser utilizados en el control poblacional en estaciones de transporte, manejo del flujo de transeúntes en una zona comercial o universidad, minería de datos en

supermercados para saber los movimientos de los clientes al interior del mismo, manejo y control de encuestas poblacionales que antes eran de manera manual, entre otras.

## **1.1 OBJETIVOS**

### **1.1.1 Objetivo General**

Evaluar el desempeño de algoritmos de sustracción de fondo orientados a conteo de personas bajo condiciones controladas

### **1.1.2 Objetivos Específicos**

- Recopilar el estado del arte en cuanto a sustracción de fondo aplicado en conteo de personas
- Seleccionar 3 algoritmos del estado del arte para la realización de las pruebas
- Definir un entorno y recopilar bases de datos para la realización de las pruebas
- Seleccionar un conjunto de características para la evaluación de los algoritmos
- Realizar pruebas de los algoritmos en videos

### **1.1.3 Objetivos Estratégicos**

- Contribuir al fortalecimiento del laboratorio de investigación en multimedia y visión (MMV) de la Universidad del Valle en el área de manipulación de video y sustracción de fondo.
- Investigar y analizar los desarrollos realizados en esta materia en el ámbito internacional.

## **1.2 PROBLEMÁTICA**

En los últimos años ha sido de gran interés el estudio del comportamiento de la movilidad y conteo de personas en diferentes ámbitos especialmente los relacionados con transporte público, flujo de peatones y personas en lugares como centros comerciales, lugares públicos e instituciones estatales [1][10].

En Colombia se presentan algunos estudios de caracterización de flujos peatonales como los realizados en el Campus Universitario de Meléndez de la Universidad del Valle [1] [10] además de implementar modelos matemáticos para dichos estudios [2] [3] [10], con el objetivo de analizar el comportamiento de los flujos peatonales en el campus y posteriormente plantear estrategias y diseños de distribución de la infraestructura que permita la optimización de los espacios.

La implementación de estos modelos matemáticos requiere que sean alimentados con datos reales tomados en campo que permitan visualizar el comportamiento del modelo para tomar decisiones. En el caso de las estaciones de transporte masivo es común ver a personas tomando datos de quienes abordan las diferentes rutas de manera manual. En estudios realizados en [1] [2] [3] y [4] fue necesario que los investigadores permanecieran un día completo en los diferentes puntos del campus registrando personas manualmente.

Han surgido algunas propuestas con relación a esta problemática, que involucran la implementación de modelos computacionales para el conteo de personas. En el proceso de

conteo de personas se deben discriminar distintos tipos de objetos que son irrelevantes para el estudio como son los autos, animales, arboles, etc.

Para resolver el problema se plantea el análisis e implementación de algoritmos y técnicas de sustracción de fondo, de los cuales se van a tener un foreground limpio de objetos que sean estáticos, detectando el movimiento en los frames, y facilitando las técnicas de conteo de personas.

Para la realización de este proyecto se requirió tanto hardware como software de aplicación. El hardware requerido constó de una cámara digital que interactúa con un computador personal en el cual se encuentra el software de procesamiento.

### **1.3 TRABAJO DESARROLLADO**

La figura 1-1 muestra un panorama general de los pasos necesarios para la realización de este trabajo. El proceso comienza con la adquisición del video, seguido de una etapa de pre procesamiento cuyo objetivo es el de facilitar las operaciones posteriores reduciendo al mínimo la pérdida de información.

Los algoritmos fueron implementados en lenguaje Python de naturaleza multiplataforma y libre licencia de desarrollo, los requerimientos mínimos recomendados para su ejecución son los siguientes:

- Plataforma(s): Windows 7, Windows 8, Windows 10, Red Hat Linux, SUSE Linux.
- Procesador: 2,1 GHz o superior
- Memoria RAM: 2 Gb o superior
- Espacio libre en disco duro: 5 Gb

Los paquetes necesarios para esta aplicación se nombran a continuación:

- Opencv 3.1
- Numpy
- Mprof

Las características recomendadas para los videos procesados, de acuerdo con las estadísticas obtenidas y para un funcionamiento exitoso son las siguientes:

- Resolución mínima de 960 x 540
- Bajo ruido de fondo
- Buenas condiciones de iluminación sin variaciones demasiado notorias

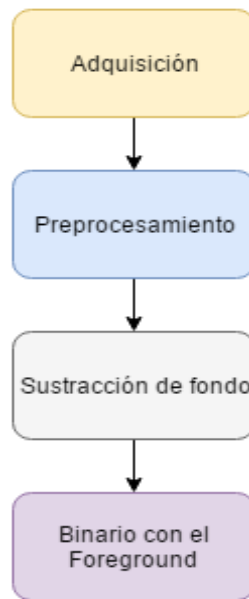


Figura 1-1 Diagrama de bloques del sistema desarrollado.

#### 1.4 BREVE INDICACIÓN DEL CONTENIDO DE CADA CAPÍTULO

Este trabajo ha sido dividido por capítulos y subcapítulos. En cada capítulo se presentan los desarrollos realizados en cada uno de los bloques mostrados en la figura 1-1 que conforman a grandes rasgos este proyecto, de forma tal que se brinde al lector una visión clara de los algoritmos y análisis presentados.

En el capítulo 2, marco conceptual, se abordan los conceptos de sustracción de fondo, los pasos presentes en gran parte de los algoritmos de este tipo, una breve mirada a los diferentes trabajos realizados a nivel nacional e internacional en el tema de sustracción de fondo; se muestran algunos ejemplos de diferentes enfoques.

En el capítulo 3, Revisión de la literatura, se detallan diferentes implementaciones en sustracción de fondo en la literatura, y se muestran trabajos realizados en la misma área pero con diferentes técnicas, se explican otras técnicas de sustracción de fondo y se cita literatura que sirvió para la construcción de este trabajo.

En el capítulo 4, Elección de algoritmos, se explican los algoritmos elegidos, las fórmulas que componen la implementación y por último se dan las conclusiones de los 3 algoritmos.

En el capítulo 5, Parámetros y criterios de comparación, se explican cuáles serán los criterios de comparación y de evaluación de los algoritmos, se da una explicación del porque las agrupaciones de los videos de prueba y se da información de los parámetros de los algoritmos.

El capítulo 6, Pruebas y Resultados, contiene los procedimientos realizados, detalles de la implementación, así como los resultados obtenidos con sus respectivos análisis.

En el capítulo 7 se presentan las conclusiones del proyecto y por ultimo las referencias bibliográficas.

# MARCO CONCEPTUAL

## 2.1 INTRODUCCIÓN

Para comprender los capítulos siguientes de este documento y el campo dentro del cual se enmarca este trabajo en la investigación mundial, resulta indispensable presentar una descripción general de las temáticas que aborda el proyecto, los antecedentes, el estado del arte y algunas consideraciones prácticas de las metodologías que se emplearon.

## 2.2 ANTECEDENTES

### 2.2.1 Antecedentes Locales

A nivel local se encuentran muy pocos trabajos relacionados con el tema, los más cercanos son segmentación pero con imágenes, encontramos uno de la Universidad Javeriana de Cali y otro en la EIEE de la Universidad del Valle:

- En la Universidad Javeriana de Cali se encuentra el trabajo **“Lector Automático de Matrículas de Automóviles a partir de Imágenes en Escala de Grises”** por los estudiantes: “Juan Pablo Marín, Julián Duque, Francisco Macías, Oscar Reina y Jairo A. Hurtado”. [Marín 2004].

Técnicas: Segmentación de caracteres utilizando histogramas y reconocimiento por medio de correlación.

- En la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle se han realizado algunos trabajos relacionados con visión artificial e inteligencia computacional, el más relacionado con este proyecto es el de **“Identificación de Matrículas Vehiculares Mediante Técnicas de Visión Artificial”** realizado por los estudiantes: Claudia Ximena Parra Garrido y Ernesto Rodríguez Collazos en el 2003. [Parra 2003].

Técnicas: Reconocimiento de caracteres empleando Redes Neuronales y propiedades del color.

### 2.2.2 Antecedentes Nacionales

A nivel nacional son también pocos los trabajos encontrados sobre el tema, estos son:

- En la Universidad Sergio Arboleda el trabajo **“Filtro mediana recursivo para la estimación de fondo y segmentación de primer plano en videos de vigilancia”** por los estudiantes: “Freddy Alexander Díaz Gonzales y David Alejandro Arévalo Suarez”. [Díaz 2014]

Técnicas: Segmentación de primer plano en video y uso de filtros.



- **“Análisis de video para estimación del movimiento humano: una revisión”**, Artículo presentado por Fabio Martínez Carrillo, Francisco Albeiro Gómez Jaramillo y Eduardo Romero Castro en la Universidad Militar Nueva Granada. [Carrillo 2009].

Técnicas: Segmentación y análisis de video para estimación de movimiento

### 2.2.3 Antecedentes Internacionales

A nivel internacional se encuentran múltiples trabajos con el objetivo de analizar y comparar algoritmos de sustracción de fondo:

- En la “Université de Sherbrooke” los investigadores “Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile y Christophe Rosenberg” realizaron el trabajo **“Comparative study of background subtraction algorithms”**. [Benezeth 2010].

Técnicas: sustracción de fondo, técnicas de comparación de algoritmos de sustracción de fondo.

- En la “University of California.” de Los Ángeles en estados unidos se realizó el trabajo **“Background subtraction on distributions”** por los investigadores: “Teresa Ko, Stefano Soatto y Deborah Estrin”. [Ko].

Técnicas: Técnicas de mejora y adaptación de algoritmos en sustracción de fondo.

Las anteriores experiencias (algoritmos utilizados, resultados positivos e inconvenientes) constituyeron una base y una ayuda muy importante para el desarrollo de este proyecto.

## 2.3 BACKGROUND SUBTRACTION

Podemos ver a lo largo de la historia que el reconocimiento de objetos en movimiento tiene un amplio conjunto de metodologías de realización, en este caso nos centramos en la sustracción de fondo, aunque existen numerosos algoritmos de sustracción de fondo en la literatura [5], la mayoría de ellos siguen un diagrama de flujos simple definido por Cheung y Kamath [5] [4], pasando por cuatro pasos generales los cuales son:

- Pre-procesamiento: las tareas simples de procesamiento de imágenes que cambia la entrada de video a un formato que pueda ser procesado en los siguientes pasos
- Modelamiento de fondo: es también conocido como mantenimiento de fondo
- Detección de primer plano: también conocido como sustracción de fondo
- Validación de los datos: referido también como post-procesamiento usado para eliminar los pixeles que no correspondan a objetos en movimiento es decir falsos positivos.

Aunque los términos sustracción de fondo y modelamiento de fondo son confundidos a menudo, estos son procesos separados y muy distintos. El modelamiento de fondo se refiere al proceso de crear, y subsecuentemente mantener, un modelo de la apariencia de fondo en el campo de vista de una cámara. La sustracción de fondo se refiere al proceso en el cual un cuadro de imagen es comparado al modelo del fondo con el fin de determinar si pixeles individuales son partes del fondo o del frente. Por ello también se le conoce como detección de primer plano.

La sustracción de fondo está basada en la suposición de que la diferencia entre la imagen de fondo y la imagen actual es causada por la presencia de objetos en movimiento. Los píxeles que no cambian son considerados como “background” o fondo y los píxeles que cambian son considerados como “foreground” u objetos en movimiento. En la práctica, los píxeles que no cambian también pueden llegar a ser parte de objetos en movimiento, por ejemplo, cuando tienen el mismo color que el fondo. De la misma manera, algunos píxeles que cambian también pueden ser clasificados como fondo cuando ocurren cambios de iluminación, así que para evitar esto se deben respetar 5 criterios:

- La cámara y sus parámetros siempre serán fijos.
- La escena no debería presentar cambios de iluminación.
- El fondo es visible y estático, es decir no hay objetos ocultos en otros objetos.
- El fondo inicial no contiene objetos del frente “dormidos” o estáticos.
- El frente y fondo pueden ser separados al hacer una comparación entre la imagen del fondo y la imagen actual, la elección de la técnica aquí es clave.

En la práctica, ninguno de estos criterios se respeta completamente debido a situaciones desafiantes que aparecen tanto en escenas interiores como en exteriores.

Actualmente en el campo de sustracción de fondo se cree que no existen soluciones perfectas. El modelamiento de fondo y la sustracción son solo puntos intermedios y no la solución definitiva. Un sistema perfecto debería estar en capacidad de resolver problemas como “bootstrapping”, “objetos en movimiento”, “movimiento de árboles”, “sombras que cambian de manera gradual o inmediata”, “camuflaje”, y demás. Algunos de estos problemas son imposibles de solucionar en paralelo pues cada uno requiere necesidades diferentes y una comprensión diferente del movimiento del primer plano y el fondo. Por lo dicho anteriormente a la hora de implementar un algoritmo de sustracción de fondo, es recomendable realizar restricciones para así centrarse en el punto fuerte de cada algoritmo y aprovechar al máximo las ventajas que nos ofrecen unas técnicas frente a otras.

## ETAPAS FUNDAMENTALES DE LA SUSTRACCIÓN DE FONDO

Esta sección explica los diferentes pasos relacionados a la sustracción de fondo, la figura 2 nos muestra una pequeña idea de su funcionamiento.

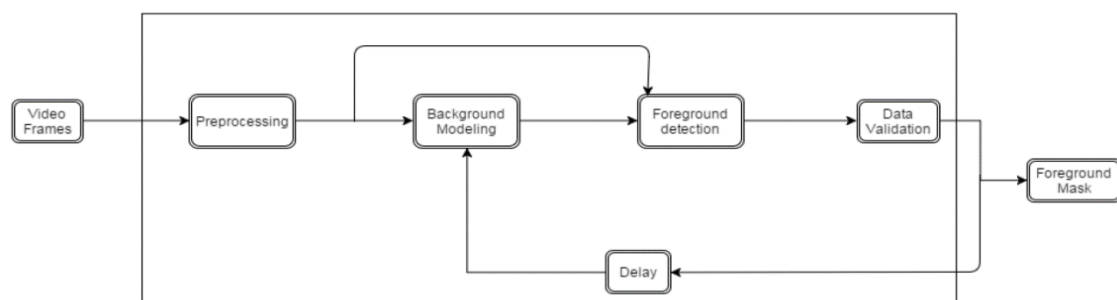


Figura 2-1 Etapas generales de sustracción de fondo.

Los algoritmos fueron implementados en Python 2.7 y a continuación se dará en detalle cada una de las etapas que componen el proceso de análisis de video.

### 2.3.1 Modelado de fondo

El modelado de fondo (también llamado representación) describe la clase de modelo usado para representar el fondo. Este esencialmente determina la habilidad del modelo para lidiar con fondos uní-modal o multi-modal.

### 2.3.2 Inicialización de fondo

La inicialización de fondo (generación, extracción o construcción) se refiere a la inicialización del modelo. Contrario a la representación del modelo de fondo y el mantenimiento del modelo, la inicialización del modelo de fondo fue investigada de manera aislada. La razón principal es que a menudo la suposición hecha es que la inicialización puede lograrse mediante la explotación de algunos frames limpios en el inicio de la secuencia. Naturalmente, esta suposición raramente es vista en escenarios reales debido a la presencia de discontinuidades. Generalmente, el modelo es inicializado usando el primer frame o un modelo de fondo sobre un conjunto de entrenamiento, los cuales pueden contener o no objetos en el fondo. El desafío principal es obtener un primer modelado del fondo cuando en el conjunto de entrenamiento, más de la mitad de frames contienen objetos de fondo. Este proceso de aprendizaje (o entrenamiento) puede ser online u off-line abriendo las puertas a las aplicaciones web con sustracción de fondo o servicios móviles que permitan la manipulación del mundo real con el virtual (realidad aumentada). Los algoritmos de inicialización dependen de la complejidad de sus modelos de fondo. Las principales investigaciones en esta área utilizan media, histogramas, tablas de intervalos y SVM [6].

### 2.3.3 Mantenimiento de fondo

El mantenimiento de fondo confía en los mecanismos usados para la adaptación del modelo a los cambios de la escena a lo largo del tiempo. El proceso del mantenimiento de fondo tiene que ser un algoritmo que sea dinámico y siempre esté conectado, pues no se sabe a ciencia cierta qué cantidad de datos serán manejados. Las fases claves de este paso son las siguientes:

- Esquemas de mantenimiento: en la literatura, se observan 3 esquemas de mantenimiento, el esquema ciego, selectivo y borroso. El mantenimiento de fondo ciego actualiza todos los pixeles con las mismas reglas las cuales son a menudo un filtro IIR

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha I_t(x, y) \quad (1.1)$$

Donde  $\alpha$  es el índice de aprendizaje el cual es una constante entre [0,1].  $B_t$  Y  $I_t$  son la imagen de fondo y la actual en el tiempo respectivamente. La desventaja principal de este esquema es que el valor de los pixeles clasificados como fondo es usado en

la computación del nuevo fondo, esto contamina la imagen de fondo y puede llegar a generar inconvenientes. Para resolver este problema, algunos autores usaron un esquema de mantenimiento selectivo que consiste en actualizar la nueva imagen de fondo con diferentes índices de aprendizaje dependiendo de una clasificación previa de los píxeles en el fondo o frente.

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha I_t(x, y) \quad (1.2)$$

*Si  $(x, y)$  es fondo*

$$B_{t+1}(x, y) = (1 - \beta)B_t(x, y) + \beta I_t(x, y) \quad (1.3)$$

*Si  $(x, y)$  es frente*

La idea aquí es adaptar muy rápidamente un pixel clasificado como fondo y muy lentamente uno clasificado como frente. Por esta razón  $\beta \ll \alpha$  y a menudo  $\beta = 0$  así que la ecuación 1.3 se resuelve:

$$B_{t+1}(x, y) = B_t(x, y) \quad (1.4)$$

Pero el problema es que una clasificación errónea puede resultar en un modelamiento de fondo incorrecto. Este problema puede ser abordado con un esquema adaptativo borroso el cual tiene en cuenta la incertidumbre de la clasificación. Esto puede conseguirse graduando la regla de actualización usando el resultado de la detección del frente tal como en [8]

- **Índice de aprendizaje:**

El intervalo de aprendizaje determina la velocidad de la adaptación de los cambios de escena. Este puede ser un valor fijo o ajustarse dinámicamente por medio de un método estadístico o Fuzzy. En el primer caso, el índice de aprendizaje es estático para la totalidad de la secuencia. Por lo tanto este puede ser determinado manualmente o puede ser seleccionado automáticamente por un algoritmo de optimización. Sin embargo, este puede tomar un valor para el aprendizaje y variar para el mantenimiento. Adicionalmente, el índice puede cambiar a lo largo del tiempo siguiendo una estrategia de retroalimentación. Para el caso Fuzzy, se usaron diferentes índices de aprendizaje por cada gaussiano en el modelo MOG mejorando así los resultados y la velocidad de convergencia de manera notable.

- **Mecanismos de mantenimiento:**

El índice de aprendizaje determina la velocidad de adaptación a los cambios de iluminación pero también el tiempo del cambio necesario en el fondo para que sea incorporado en el modelo así como el tiempo que un objeto estático del frente puede sobrevivir, es decir, sea incluido en el modelo. Así, el índice de aprendizaje lidia con diferentes desafíos los cuales tienen características temporales diferentes. Para

desacoplar el mecanismo de adaptación y el mecanismo de incorporación, algunos autores, usaron un conjunto de contadores los cuales representan la cantidad de veces que un pixel se clasifica como pixel de frente. Cuando este número es más grande que una cota fijada, el pixel es considerado como un pixel estático del frente. Otro método diseñado por lindstrom [6] usa un detector CUSUM para determinar cuando los componentes deberían ser transferidos del frente al fondo. El detector CUSUM utiliza contadores locales para los componentes del frente. Si el contador excede un límite, el componente de primer plano correspondiente es añadido al modelo del fondo. Si el número máximo de los componentes de fondo fue alcanzado, este reemplaza el menor. El componente del frente permanece en el modelo porque otro pixel podría ser descrito por este.

- **Frecuencia de la actualización:**

El objetivo es actualizar únicamente cuando sea necesario. El mantenimiento puede hacerse en cada cuadro pero si no se presentan cambios bruscos, no es necesario actualizar todos los pixeles en cada cuadro. Por ejemplo, Porikli [6] propuso un periodo de adaptación del tiempo del mecanismo de mantenimiento con respecto a un cambio de puntaje en la iluminación. La idea es que no se aplique un mantenimiento si no se detectan cambios en la iluminación, de lo contrario sería necesario realizarlo.

#### 2.3.4 Detección del primer plano

La detección del frente o primer plano consiste en comparar la imagen de fondo con la imagen actual para clasificar pixeles como de fondo o de frente. Esta tarea es una clasificación que puede ser lograda por herramientas de clasificación estadísticas, crisp (frescas) o fuzzy. Para esto, los diferentes pasos son presentados:

- **Pre procesamiento:**

Los pasos de pre procesamiento evitan la detección de cambios insignificantes debido al movimiento de la cámara o cambios de iluminación. Este paso puede involucrar ajustes geométricos y de intensidad. Los ajustes geométricos pueden ser desarrollados usando transformaciones espaciales de bajas dimensiones tales como similaridad, affine o transformaciones proyectivas. Por otro lado, hay varias maneras de lograr ajustes de intensidad. Estos pueden ser logrados con normalización de intensidad. Los valores de intensidad de los pixeles en la imagen actual son normalizados para tener la misma media y varianza que los de la imagen de fondo. Otra manera consiste en usar un filtro homomorfo el cual está basado en modelado de sombras. Esta solución permite separar la iluminación y la reflectancia. Pues solo los objetos con reflectancia contienen información de los objetos en la escena.

- **Pruebas:**

las pruebas que permiten clasificar pixeles de la imagen actual como fondo o frente son generalmente la diferencia entre la imagen de fondo y la imagen actual. Esta

diferencia es umbralizada. otras formas de comparar dos imágenes son pruebas de significado e hipótesis. La regla de decisión es llamada como una prueba de hipótesis estadística. La decisión de si ocurrió o no un cambio en cada pixel corresponde a elegir uno o dos hipótesis: la hipótesis  $H_0$  o una hipótesis alternativa  $H_1$ , correspondiente a decisiones de cambio o no cambio. Varias pruebas de significado pueden encontrarse en [11][12][13][14][15][16][17][18]

- **Threshold:**

En la literatura, hay varios tipos de esquemas de umbral. Primero, el umbral puede ser fijo y estático para todos los pixels y la secuencia. Este esquema es simple pero no es óptimo. En efecto, los pixels presentan diferentes actividades y necesitan un umbral que se adapte. Este puede realizar al computar el umbral por medio de la desviación estándar temporal de intensidad entre el fondo y las imágenes actuales, y al actualizarlo usando un filtro IRR. Un umbral adaptable puede ser obtenido estadísticamente de la varianza del pixel, otra manera de adaptación de umbral es usar umbrales Fuzzy.

- **Pos procesamiento:**

La idea aquí es mejorar la consistencia de la máscara de frente. Esto puede lograrse al borrar pixels aislados con operadores morfológicos estadísticos. Otra forma es usar conceptos fuzzy tales como interferencia fuzzy entre la máscara anterior y la actual.

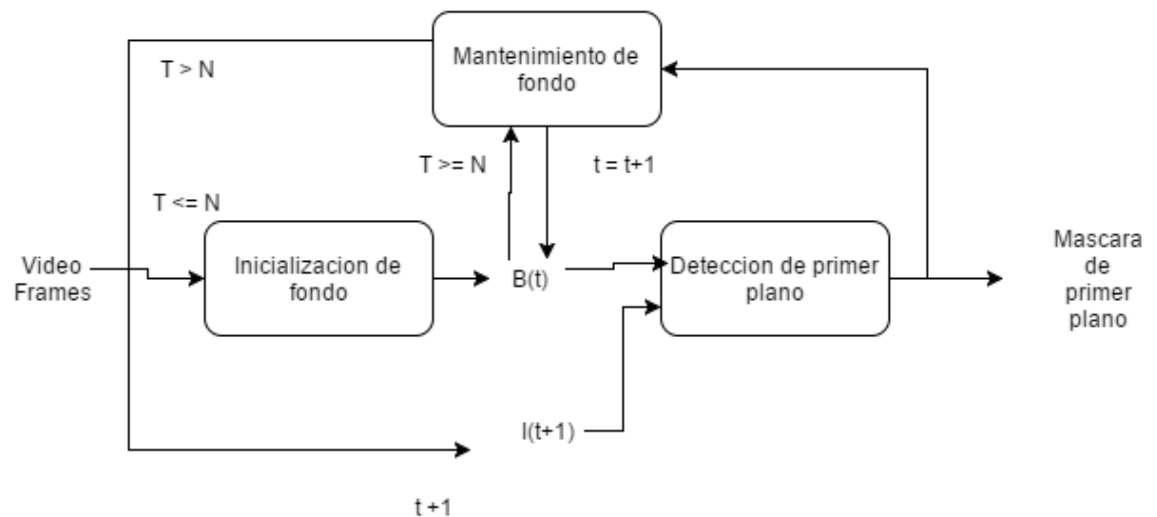


Figura 2-2 proceso de sustracción de fondo  $N$  es el número de frames que son usados para iniciar el fondo  $B_t$  y  $I_t$  son el fondo y la imagen actual en el tiempo  $t$  respectivamente

## 2.4 CONCLUSIONES

Actualmente los algoritmos de sustracción de fondo están siendo ampliamente utilizados en las nuevas tecnologías que están apareciendo, debido a su naturaleza de tiempo real, encontramos que trabaja en la captura de los movimientos del Kinect, y seguramente será bastante empleada con las futuras tecnologías emergentes como los google glasses que son gafas de realidad aumentada donde por medio de unas pequeñas modificaciones de los algoritmos para hacer algunas operaciones en gpu, podemos lograr un rendimiento muchas veces mayor a que si ejecutáramos simplemente en CPU.

# REVISIÓN DE LA LITERATURA

### 3.1 INTRODUCCIÓN

En este capítulo agrupados por secciones los diferentes artículos que soportaron la investigación, los criterios de separación de los diferentes artículos son por técnica empleada, es decir, separaremos en 3 secciones principales el contenido de este capítulo, la primera son algoritmos tradicionales de sustracción de fondo donde se encuentran los algoritmos clásicos para hacer segmentación, en la segunda sección tenemos algunas implementaciones recientes de sustracción de fondo, y por último la sección de implementaciones en tiempo real que se va a centrar en implementaciones de GPU, no se consideraron las implementaciones embebidas, de arquitectura o paralelas para esta sección.

### 3.2 MODELADO DE FONDO TRADICIONAL

#### Modelos estadísticos:

Los modelos estadísticos nos ofrecen una robustez a cambios de iluminación considerable y a cambios en el fondo. Los modelos estadísticos pueden ser clasificados en estas categorías;

- **Modelos gaussianos:** La manera más simple de representar el fondo es asumir que la intensidad de los píxeles en el tiempo pueden ser modelados por un gaussiano. Siguiendo esta idea, en [27] se propuso usar un solo gaussiano (conocido como SG). En [28] se generalizó el SG usando un solo gaussiano general (SGG) para aliviar las restricciones estrictas de un gaussiano. Sin embargo, un modelo unimodal no puede lidiar con fondos dinámicos cuando hay ondeo de árboles, ondulación de agua o algas moviéndose. Para resolver este problema, la mezcla de gaussianos (MOG) o modelo de mezcla de gaussianos (GMM) presentado en este trabajo se comporta muy bien frente a fondos dinámicos.
- **Modelos soportados en vectores:** La segunda categoría usa modelos estadísticos más sofisticados como SVM (Support Vector Machine) [29], SVR (Support vector regression) [30] y SVDD (Support Vector Data Description) [31]. Más adelante se propone inicializar el fondo usando un SVM (Support vector machine). La clasificación SVM se aplica a cada píxel en los frames de entrenamiento al calcular la probabilidad de los resultados. Los píxeles nuevos son evaluados y se determina si se agregan al modelo de fondo. la inicialización de fondo continúa hasta que no hay más píxeles de fondo a considerar. De manera similar en [29] [32] se usó un SVR diferente para modelar cada píxel del fondo como función de la intensidad. La inicialización de fondo se hace usando un algoritmo batch y el mantenimiento se logra con un algoritmo SVR online.
- **Modelos de aprendizaje de subespacio:** La tercera categoría emplea el aprendizaje de subespacio. El aprendizaje de subespacio usando componentes principales de



análisis (SL-PCA) [33] se aplica a  $N$  imágenes para construir un modelado de fondo, el cual es presentado por una imagen media y la matriz de proyección involucrando los primeros  $p$  vectores propios que sean significantes de PCA. De esta forma, la segmentación del frente se completa al calcular la diferencia entre la imagen de entrada y su reconstrucción. Sin embargo, este modelo presenta muchas limitaciones[34]

### Modelos de redes neuronales:

En este caso, el fondo se representa por la media de los pesos de una red neuronal entrenada adecuadamente en  $N$  frames limpios. La red aprende como clasificar cada pixel como fondo y frente. Típicamente, las neuronas de la red codifican el modelo de fondo de un pixel del frame completo. En el contexto de sustracción de fondo, la aplicación de tipos particulares de redes neuronales se ha vuelto beneficiosa pues permite paralelización intrínseca. Esta propiedad de las redes neuronales permite implementaciones eficientes en hardware. Las principales propuestas se pueden clasificar de la siguiente forma:

- **Red neuronal general de regresión:** Culibrk [35] propuso usar la arquitectura de una red neuronal para formar un clasificador bayesiano para la detección de fondo y frente. El clasificador construido maneja eficientemente la segmentación de secuencias de escenas naturales con movimientos complejos de fondo y cambios de iluminación. Los pesos permiten modelar el fondo y son actualizados para reflejar los cambios estadísticos del fondo. además, esta propuesta es paralelizable a nivel de sub pixeles y se puede mejorar su eficiencia en hardware.
- **Redes neuronales multivaluadas:** [36] uso un método de detección de frente basado en el uso de redes neuronales multivaluadas discretas. El modelo de red multivaluada es usado para detectar y corregir algunas deficiencias y errores del algoritmo MOG. Una de las ventajas de usar redes neuronales multivaluadas para detección de frente es que todas las unidades del proceso (neuronas) computan la solución al problema en paralelo. Otra ventaja de modelos de redes multivaluadas es su habilidad para representar clases no numéricas o estados, los cuales pueden ser muy útiles cuando se lidia con problemas de detección de frente, en los cuales los estados de los pixeles son definidos con capas cualitativas: frente, fondo y sombras.
- **Redes neuronales competitivas:** En [37] se usa una red neuronal competitiva sin supervisión (CNN) basado en vecindades adaptables para construir modelos del fondo. los pesos y las vecindades adaptables de las neuronas modelan el fondo y se actualizan. Este algoritmo es paralelizado a nivel pixel y diseñado para implementaciones en hardware para trabajar en tiempo real.

### 3.3 MODELADO DE FONDO RECIENTE

La representación de modelos de fondo recientes investigados para este trabajo se puede clasificar en las siguientes categorías: modelos estadísticos avanzados y modelos Fuzzy, estos a su vez tienen unas clasificaciones.

#### **Modelos estadísticos avanzados:**

Los modelados estadísticos avanzados de fondo se desarrollaron hace poco y según la literatura se clasifican en los siguientes grupos:

- **Modelos de mezcla:** En esta categoría, los autores usaron una distribución diferente a la gaussiana usada en GMM. Algunos autores usaron el Student-t Mixture model [38] [39] o el modelo Dirichlet Mixture [40] [41]. El modelo Student's t-mixture (STTM) ha probado ser muy robusto contra ruidos debido a su naturaleza comparado a los modelos de mezclas gaussianos[40] pero STTM no se había aplicado anteriormente a procesamiento de video, pues los algoritmos EM no pueden ser aplicados en el proceso. El gran incremento en complejidad haría imposible implementaciones de tiempo real, aunque en [40] se propone un nuevo filtro recursivo en tiempo real para actualizar los parámetros de la distribución de manera efectiva. así que, el método permite modelar el fondo y luego separar el frente con gran precisión en el caso de objetos de fondo lentos y fondos dinámicos.
- **Modelos híbridos:** en esta categoría tenemos la implementación de [42] en la cual uso un modelo de mezcla paramétrico no regional (KDE) y un modelo paramétrico de pixeles (GMM) para aproximar la distribución de color del fondo. la distribución de color del fondo es entrenada de pixeles vecinos del frame anterior. Las distribuciones locales del fondo y frente son aproximadas con un modelo no paramétrico (KDE). La coherencia temporal se modela con cadenas de markov. así, color, localización, coherencia temporal y consistencia son puestas juntas en el mismo framework. Los modelos de color, localidad y coherencia temporal se entrenan online de fondos dinámicos complejos. Siguiendo la misma idea [43] [44] usaron un modelo híbrido entre KDE y GMM. Siguiendo este framework probabilístico, este método lidia con detección de fondo y remoción de sombras, simultáneamente al construir las funciones de probabilidad de densidad (PDFs) de objetos en movimiento y estáticos, aquí estos PDFs se construyen basados en el modelo híbrido KDE-GMM el cual tiene las fortalezas de ambas implementaciones.
- **Modelos no paramétricos:** tenemos en esta implementación propuesta en [45] en la cual tenemos un algoritmo llamado Visual Background extractor (ViBe) que construye el modelo de fondo al agregar valores observados anteriormente de la posición de los pixeles. La clave de ViBe es una política de selección aleatoria que asegura que su crecimiento exponencial sea suave. La segunda innovación involucra el post procesamiento para darle consistencia espacial al usar un método de propagación rápido y que contiene información espacial que de manera aleatoria difunde los valores de los pixeles a lo largo de sus pixeles vecinos. La tercera innovación está relacionada a la inicialización de fondo la cual es instantánea y permite al algoritmo iniciar a partir del segundo frame de la secuencia. Aunque ViBe muestra unos resultados aceptables en muchos escenarios, es problemático con escenarios desafiantes como fondos oscuros, sombras, y cambios frecuentes en el fondo.

## Modelos Fuzzy:

Todas las situaciones críticas generan imprecisiones e incertidumbres en el proceso completo de la sustracción de fondo. Además, algunos autores recientemente introducen el concepto de Fuzzy en los diferentes pasos de la sustracción de fondo para tener en cuenta estas imprecisiones e incertidumbres. En [46] se da una clasificación reciente de este tipo de métodos:

- **Modelado de fondo Fuzzy:** el desafío principal aquí consiste de modelar fondos multi modales. El algoritmo más usado es el GMM para lidiar con este desafío pero los parámetros son determinados una secuencia de entrenamiento que contiene datos insuficientes o con ruido. De modo que, los parámetros no son determinados muy bien. En este contexto, se utilizan modelos T2F-MOG (Type-2 Fuzzy Mixture of Gaussians) para modelar ciertas incertidumbres cuando surgen fondos dinámicos. Para resolver estos inconvenientes hay 2 implementaciones propuestas, la primera es para la incertidumbre sobre la media y otro sobre la varianza, llamados T2-FMOG-UM y T2-FMOG-UV, respectivamente. El T2-FMOG-UM y T2-FMOG-UV son más robustos que el MOG. Prácticamente, T2-MOG-UM es más robusto que T2-FMOG-UV. En efecto, las medias se estiman a medida que pasa el tiempo en el mantenimiento del MOG. La varianza y los pesos son inestables y poco confiables, así que genera menos robustez en T2-FMOG-UV.
- **Mantenimiento Fuzzy de fondo:** La idea es actualizar el fondo siguiendo el registro que dejan los pixeles al ser clasificados como fondo o frente. El registro viene de la detección de frente Fuzzy y se introduce en el esquema de mantenimiento de dos maneras. La primera manera [47] consiste en adaptar de manera Fuzzy el aprendizaje siguiendo la clasificación del pixel. La segunda manera implica que las reglas del mantenimiento se vuelven una combinación Fuzzy de dos reglas divididas [48] [49]. La adaptación del mantenimiento de fondo Fuzzy permite lidiar robustamente con cambios de iluminación y sombras.

## 3.4 IMPLEMENTACIONES EN TIEMPO REAL

La sustracción de fondo generalmente se hace en el contexto de aplicaciones de tiempo real en computadores ordinarios o en cámaras embebidas. Así que, es necesario lograr un balance entre el tiempo de computación y los requerimientos de memoria y tratar de mantener un equilibrio entre esas dos. Lo poco investigado sobre estas implementaciones nos lleva a implementaciones en tiempo real en GPU, la cual en muchos casos no es usada correctamente en los equipos y se hace abuso del CPU en su lugar desaprovechando una gran capacidad de procesamiento y paralelismo por ignorancia sobre ello. En [50] y [25] se proponen implementaciones de tiempo real usando el lenguaje CUDA para el SG (single gaussian), el codebook y el SOBS. La mayor parte de la literatura se enfoca sobre el algoritmo MOG, pero también se encuentran algunas implementaciones en SVM [51] [52] [53]

### 3.5 CONCLUSIONES

Tenemos bastantes tipos de algoritmos cuando se habla de sustracción de fondo y muchos tipos de implementaciones, siendo las estadísticas las más utilizadas y de las que más existe documentación, también recalcar que se encuentran muchas variantes de los algoritmos GMM y MOG y muchas opciones de mejoramiento de los problemas encontrados comúnmente como el bootstrapping o el camuflaje.

# ELECCIÓN DE ALGORITMOS

## 4.1 INTRODUCCIÓN

En este capítulo se describen los algoritmos que serán implementados y comparados, sus diferentes fórmulas y consideraciones matemáticas y un esquema general del funcionamiento del mismo, el algoritmo static frame difference fue elegido debido a su naturaleza intuitiva, es decir, es donde se aprecia de manera más sencilla, las bases de la sustracción de fondo sin ningún tipo de mejora frente a aspectos adversos, el KNN y el MOG fueron elegidos por el tipo de algoritmos que son, la documentación que existe sobre ellos y recomendación de otros autores.

Se da una breve descripción de las características que hacen especial el algoritmo frente a otros y al final una conclusión general frente a lo explicado más adelante sobre los métodos.

### Presentación de los algoritmos utilizados.

A continuación se da una descripción de los algoritmos utilizados y algunas características de los mismos así como una descripción de tallada de los procesos matemáticos.

## 4.2 STATIC FRAME DIFFERENCE

El método de Static Frame difference o diferencia de frames estáticos es conocido también como la diferencia de frames adyacentes, el método de la diferencia de secuencia de imagen, etc. Este se refiere a unos intervalos de tiempo muy pequeños  $\Delta t$  de las dos imágenes antes y después de la diferencia en tiempo de los píxeles, y luego umbralizando para extraer las regiones de la imagen del movimiento, de acuerdo a cuales cambios dados en la diferencia como se muestra en la figura 3-4.

El método específico de cálculo de diferencia de imagen  $D_k$  entre la  $K$ -ésima imagen del frame  $f_k$  con el  $k-1$  la imagen del frame  $f_{k-1}$  es diferencial, el diferencial negativo y el diferencial completo, la fórmula correspondiente es la siguiente.

Diferencial

$$D_k = \begin{cases} f_k - f_{k-1} & \text{si } (f_k - f_{k-1}) > 0 \\ 0 & \text{de lo contrario} \end{cases} \quad (3.1)$$

Diferencial negativo

$$D_k = \begin{cases} |f_k - f_{k-1}| & \text{si } (f_k - f_{k-1}) < 0 \\ 0 & \text{de lo contrario} \end{cases} \quad (3.2)$$

Diferencial completo

$$D_k = |f_k - f_{k-1}| \quad (3.3)$$

Al finalizar la aplicación de los diferentes métodos se puede conseguir una colección de píxeles en movimiento al realizar la binarización del resultado.

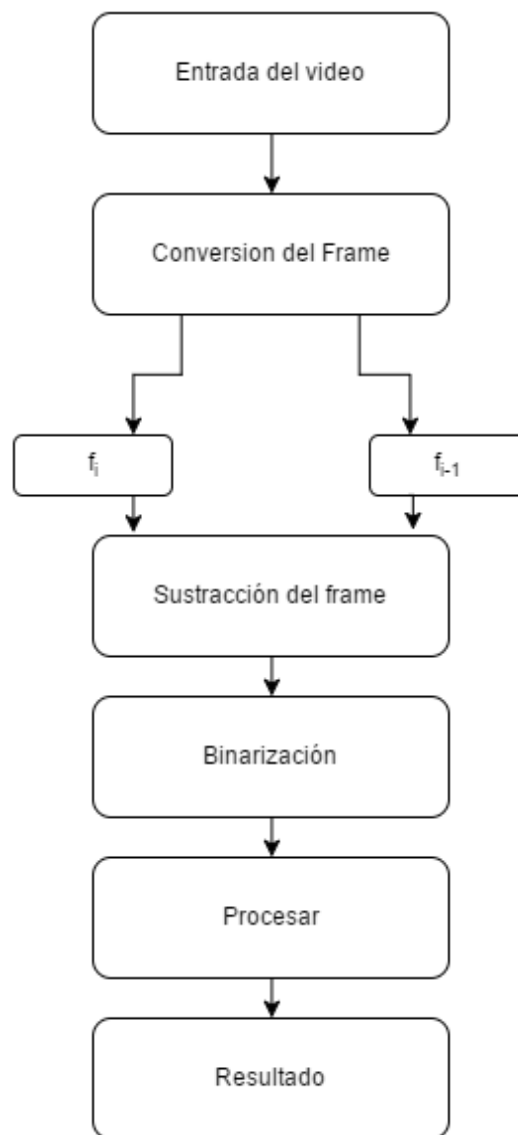


Figura 4-1. Flujo de un algoritmo de diferencia de frames

### 4.3 KERNEL DENSITY ESTIMATION

La estimación de densidad usando un kernel uniforme comienza al contar el número de muestras  $k$  del data set  $\mathbf{x}_T$  que está dentro del volumen  $V$  del kernel. El volumen  $V$  es una hipersfera de diámetro  $D$ . la densidad estimada es dada por

$$p \text{ no parametric } (\vec{X} | \mathbf{x}_T, BG + FG) = \frac{1}{TV} \sum_{m=t-T}^t K \left( \frac{\|\vec{x}^{(m)} - \vec{x}\|}{D} \right) = \frac{k}{TV} \quad (3.4)$$

Donde la función del kernel  $K(u) = 1$  si  $u < \frac{1}{2}$  y 0 de otra forma. El volumen  $V$  del kernel es proporcional a  $D^d$  donde  $d$  es la dimensionalidad del dato. Otras funciones del kernel más sencillas son usadas a menudo. En la práctica el kernel de  $K$  tiene poca influencia pero la elección de  $D$  es crítica.

La estimación de kernel consiste en el uso de un kernel fijo de tamaño  $D$  para la función de densidad la cual puede no ser la mejor elección. El también llamado “estimador de globo” adapta el tamaño del kernel por cada punto nuevo  $X$  hasta que una cantidad fija de datos es analizada. De estas formas obtenemos grandes áreas con pequeños números de ejemplos y kernels más pequeños en las áreas de población densa. La estimación del globo es usada a menudo para problemas de clasificación pues es relacionada a la clasificación K-NN. El uso de un solo vecino es usado pero para ser más robustos usamos  $K = [0.1T]$  donde  $[\cdot]$  es el operador de integral redondeado.

La implementación de este algoritmo es basada en una gaussiana pero utilizando el kernel y el concepto del vecino más cercano. Según la literatura, este algoritmo se comporta de manera óptima en casos en que hay pocos pixeles en el frente, es decir escenas de poco movimiento.

### 4.4 MOG

Es un algoritmo de segmentación de fondo y frente basado en mezcla gaussiana, una característica destacada de este algoritmo es que selecciona el numero apropiado de distribuciones gaussianas para cada pixel. Provee una mejor adaptación a escenas cambiantes debido a iluminación etc.

En la práctica, la iluminación en escena podría cambiar gradualmente (condiciones de día o climáticas en una escena exterior) o de repente (apagando la luz en una escena interior). Se puede o remover de la escena un objeto. Con el fin de adaptarse a cambios podemos actualizar el conjunto de entrenamiento al agregar nuevas muestras y descartar las viejas. Elegimos un periodo de tiempo razonable  $T$  y en el tiempo  $t$  tenemos  $\mathbf{X}_T = \{\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-T)}\}$ . Por cada muestra nueva, actualizamos el conjunto de entrenamiento  $\mathbf{X}_T$  y volvemos a calcular  $\hat{p}(\vec{X} | \mathbf{x}_T, BG + FG)$ . Sin embargo, entre las muestras recientes podría haber valores que pertenezcan a objetos del frente y deberemos denotar este estimado como  $\hat{p}(\vec{X}^{(t)} | \mathbf{x}_T, BG + FG)$ . Usamos GMM con  $M$  componentes:

$$\hat{p}(\vec{X} | \mathbf{x}_T, BG + FG) = \sum_{m=1}^M \hat{\pi}_m N(\vec{x}; \hat{\mu}_m, \hat{\sigma}_m^2 I) \quad (3.5).$$

Donde  $\hat{\mu}_1, \dots, \hat{\mu}_M$  son las medias estimadas y  $\hat{\sigma}_1, \dots, \hat{\sigma}_M$  son los estimados de las varianzas que describen los componentes gaussianos. Las matrices de covarianza se supone que son diagonales y la matriz de identidad  $I$  tiene dimensiones apropiadas. La mezcla de los pesos denotada por  $\hat{\pi}_m$  son no negativos y se incrementan hasta uno. Dado una nueva muestra  $\vec{X}^{(t)}$  en el tiempo  $t$  la ecuación de la actualización recursiva está dada por:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha (o_m^{(t)} - \hat{\pi}_m) \quad (3.6)$$

$$\hat{\mu}_m \leftarrow \hat{\mu}_m + o_m^{(t)} (\alpha / \hat{\pi}_m) \hat{\delta}_m \quad (3.7)$$

$$\hat{\sigma}^2 \leftarrow \hat{\sigma}^2 + o_m^{(t)} (\alpha / \hat{\pi}_m) (\hat{\delta}_m^T \hat{\delta}_m - \hat{\sigma}_m^2) \quad (3.8)$$

Cuando  $\hat{\delta}_m = \vec{x}^{(t)} - \hat{\mu}_m$ . En lugar del intervalo de tiempo  $T$  que fue mencionado arriba, aquí la constante  $\alpha$  describe un exponencial de cayente que es usado para limitar la influencia de datos antiguos. Mantenemos la misma notación teniendo en cuenta que aproximadamente  $\alpha = 1/T$ . Para un ejemplo nuevo a la propiedad  $o_m^{(t)}$  se le asigna 1 para el componente “cerrado” con el mayor  $\hat{\pi}_m$  y a los otros se les asigna 0. Definimos que una muestra está “cerrada” a un componente si la distancia Mahalanobis del componente es por ejemplo menor que tres desviaciones estándar. La raíz de la distancia del  $m$ -avo componente es calculada como:  $D_m^2(\vec{x}^{(t)}) = \hat{\delta}_m^T \hat{\delta}_m / \hat{\sigma}_m^2$ . Si no hay un componente “cerrado” se genera un componente nuevo con  $\hat{\pi}_{M+1} = \alpha, \hat{\mu}_{M+1} = \vec{x}^{(t)}$  y  $\sigma_{M+1} = \sigma_0$  donde  $\sigma_0$  es una varianza inicial apropiada. Si se llega al número máximo de componentes descartamos los componentes con el menor  $\hat{\pi}_m$ .

## 4.5 CONCLUSIONES

Los algoritmos presentados son robustos y capaces en varias situaciones pero como se mencionó antes, ninguno es capaz de considerar la totalidad de desafíos que se presentan en escenarios reales. Por tanto en las pruebas se van a someter a diferentes situaciones y a evaluar y consignar las situaciones de mejor y peor desempeño de los algoritmos, cabe resaltar que la explicación de los algoritmos de este capítulo es bastante sencilla y no tiene en cuenta la totalidad de las técnicas que existen para mejorar la robustez de los métodos frente a diversas situaciones. Señalar también que hay muchas variantes de los algoritmos arriba descritos y que la comunidad sigue en la búsqueda de soluciones que permitan afrontar los desafíos que tras cada avance tecnológico surgen, como el manejo de un video en resolución 4k en tiempo real o la misma realidad aumentada.



## Capítulo 5

# PARAMETROS Y CRITERIOS DE COMPARACIÓN

### 5.1 INTRODUCCIÓN

En este capítulo se introducen los videos a utilizar, los parámetros de cada algoritmo utilizado, las condiciones y agrupaciones en las que se clasificaron los videos y el porqué de estas clasificaciones, así como los criterios de comparación que posteriormente serán utilizados en las pruebas.

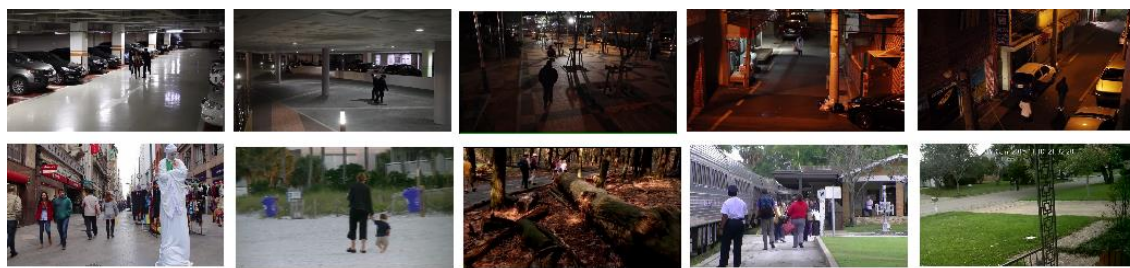


Figura 5-1. Conjunto de videos utilizados en las pruebas

### 5.2 CONJUNTO DE VIDEOS DISPONIBLES

Los videos fueron tomados de diferentes fuentes, la principal de ellas fue un servidor privado de Amazon y la página web de veezy en donde se encuentran videos gratuitos y de pago, los videos fueron clasificados en varios grupos que se describen a continuación:

#### Primer conjunto de videos:

En este conjunto encontramos los video clips de la figura 4-2, estos videos se caracterizan por estar todos grabados de noche, tienen una baja densidad poblacional por lo que el principal fuerte de estos es evaluar el desempeño de los algoritmos con baja iluminación. Los videos presentan situaciones de robos simuladas en la vida real, en la que se sobreponen las personas en el video por lo tanto también se evalúa la capacidad de diferenciar un objeto de otro en las condiciones nombras anteriormente.

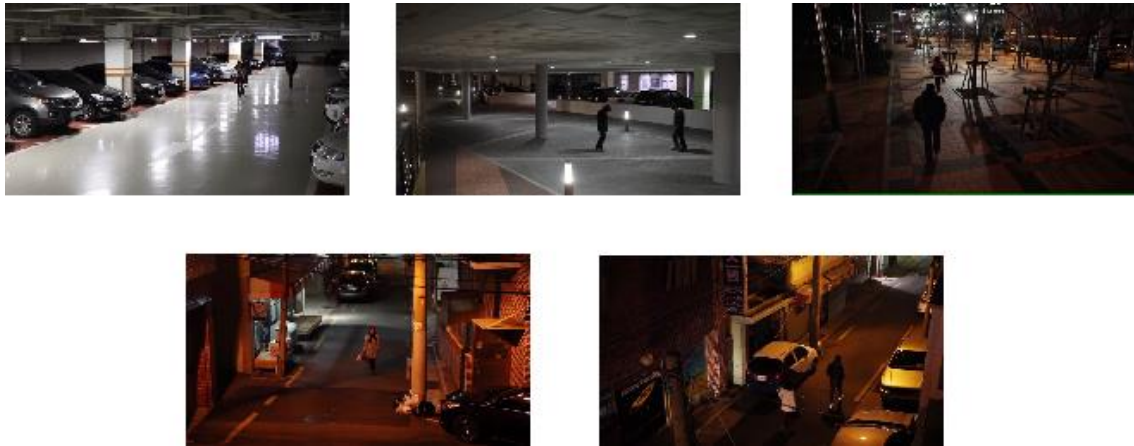


Figura 5-2. Videos nocturnos y de baja iluminación.

### Segundo conjunto de videos:

En este segundo conjunto de videos encontramos uno con condiciones diferentes a los anteriores, nos encontramos con una situación real de una esquina con alto movimiento, es decir, pixeles con información relevante, en el caso de este video se evalúa la capacidad del algoritmo de manejar altas densidades poblacionales, condiciones de iluminación normales, la cámara fija completamente y la capacidad de diferenciar masas de gente que camina en conjunto.



Figura 5-3. Videos con alta densidad poblacional.

### Tercer conjunto de videos:

En esta ocasión tenemos videos de tres situaciones diferentes, el primer clip es un video en el cual la cámara se mueve por ser sostenida por una persona, la calidad del video decayó debido a la iluminación así que este también evalúa eso de los algoritmos, en el segundo clip tenemos el de mayor desafío de todo el dataset, en esta oportunidad tenemos un video de una persona caminando en una excursión, esto implica un fondo completamente distinto en cada frame a diferencia del anterior que había fragmentos parecidos, condiciones malas de iluminación pues se encuentran al interior de un bosque, y una mala calidad del video, hacen del clip 8 el video más desafiante al que se tengan que enfrentar estos algoritmos en las pruebas. Por ultimo tenemos otra situación muy común que es el zoom de la cámara, reduciendo así calidad y cambiando las dimensiones del fondo, evaluando como en el clip 6 la sobre posición de personas y la adaptación a cambios repentinos y no transicionales del fondo.



Figura 5-4. Videos sin cámara estática

#### Cuarto conjunto de videos:

En esta ocasión tenemos otro de los inconvenientes comúnmente encontrados por los algoritmos de sustracción de fondo, y son pequeños cambios en el fondo, en este caso el ondeo de las hojas de los árboles, al mismo tiempo que se evalúa que tan bien modelan el fondo cuando la distancia de las personas a analizar es bastante superior a la encontrada en los clips anteriores.



Figura 5-5 Video con Ondeo de arboles

### 5.3 CRITERIOS DE COMPARACIÓN

Los algoritmos serán calificados de acuerdo a 2 criterios cada uno, el primer criterio de comparación y por ende las primeras pruebas a realizar, son de consumo de memoria y de tiempo, este criterio fue elegido debido a que en la realidad, los algoritmos deben poder manejar situaciones de manera instantánea, es decir, en tiempo real. El segundo criterio de comparación será la capacidad de adaptación de los algoritmos y de precisión en la detección de movimiento, esto será evaluado como se mencionó en los videos de arriba con diferentes escenarios como por ejemplo mala iluminación, movimiento del fondo, sobre posición de personas, etc. En este caso se tiene en cuenta la precisión y la calidad de la máscara entregada por cada algoritmo.

Para resumir se hará un equilibrio entre precisión y calidad de la máscara entregada y el consumo de memoria y tiempo de cada algoritmo por separado, por último en las conclusiones se da unas recomendaciones de uso de cada algoritmo según el resultado de las pruebas.

## 5.4 PARAMETROS DE LOS ALGORITMOS

**Frame difference:** Para este algoritmo no se tiene en cuenta un conjunto de entrenamiento de imágenes previas del video, sino que para mejorar su adaptación a cambios se aplica una variante del algoritmo de frame difference que evalúa el frame anterior al actual, adaptándose así mejor a situaciones adversas a cambio de precisión.

**Knn:** En este algoritmo tenemos activada la detección de sombras por defecto para obtener una máscara rica en información los demás detalles del algoritmo fueron explicados en el capítulo 3.

**MOG:** En este caso al igual que en KNN tenemos la detección de sombras sacrificando velocidad de procesamiento pero mejorando la información de la máscara, el número de distribuciones gaussianas es asignado automáticamente para cada pixel por el algoritmo, mejorando así la adaptación a escenas con variaciones de iluminación, etc.

# PRUEBAS Y RESULTADOS

## 6.1 INTRODUCCIÓN

En este capítulo se consignan las pruebas realizadas sobre los algoritmos, sobre los videos introducidos en el capítulo 5, así como también detalles de la implementación como el pre procesamiento de los videos y la librería usada para las pruebas.

Las pruebas se separaron en 2 etapas, la primera etapa mide la memoria en función del tiempo sobre cada uno de los videos para los 3 algoritmos y se presentan en un gráfico los resultados y un análisis por conjunto de clips, la segunda etapa por otro lado nos muestra los gráficos de precisión y recall que miden la precisión de los algoritmos en la detección efectiva de movimiento.

## 6.2 PREPROCESAMIENTO DE LOS VIDEOS

En esta etapa se realiza la adecuación de los videos para su posterior procesamiento. La adecuación primero que todo consiste en el cambio de resolución de los videos para obtener resultados en las mismas condiciones, la resolución trabajada fue de 960 x 540 pixeles. Esta etapa se puede resumir en el cambio de tamaño de los videos y en la aplicación de filtros para reducir sus niveles de ruido, como se ilustra en el diagrama de flujo de la figura 3-3.

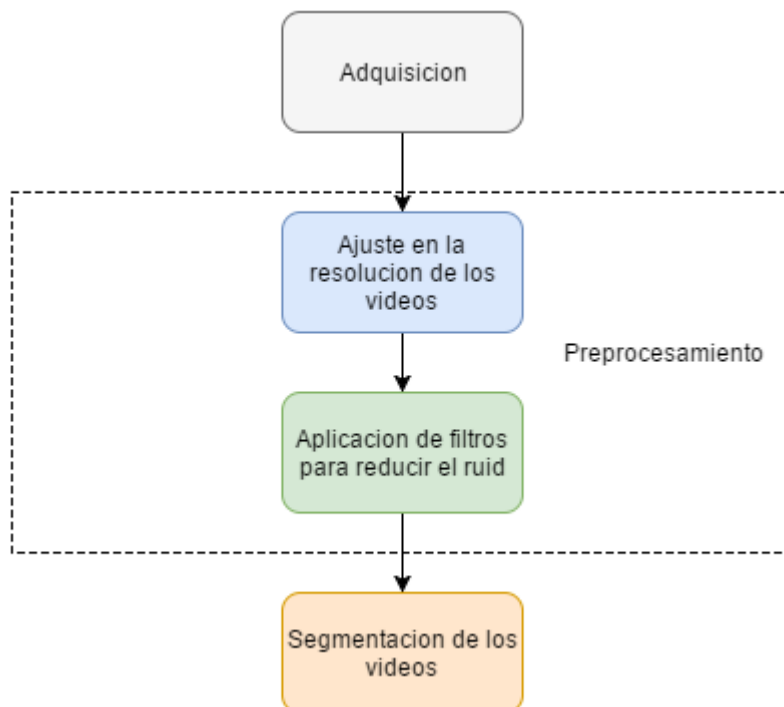


Figura 6-1. Diagrama de flujo Pre procesamiento del video

### 6.2.1 Cambiar resolución

En esta etapa se utilizó el software HandBrake para cambiar la resolución de todos los videos a 960 x 540 de manera que se obtengan resultados similares en los videos con condiciones valga la redundancia similares.

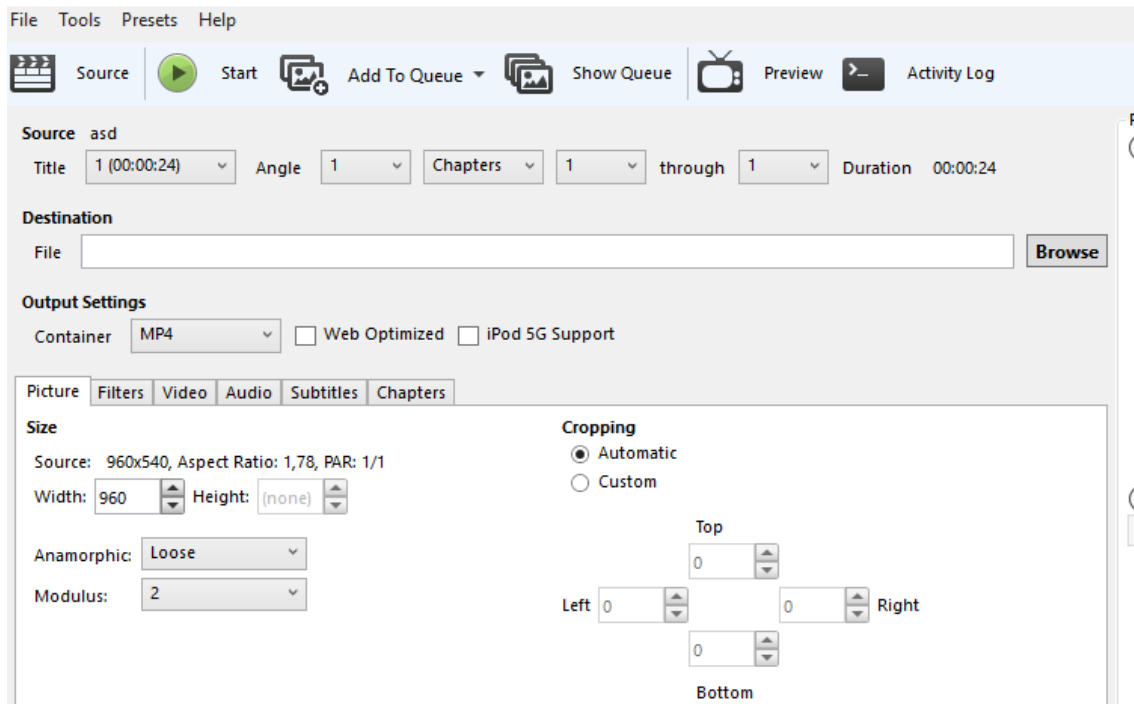


Figura 6-2. Proceso de modificación de resolución

### 6.2.2 Aplicación de filtros

Se utilizaron filtros de software de terceros como Sony vegas que es un programa para edición de videos en el cual podemos reducir el ruido de fondo por medio de un plug in del programa, este proceso se realizó con los 10 videos y no hubo ningún inconveniente notando una gran mejoría en la calidad dela imagen perdida a la hora de bajar o aumentar resolución.

## 6.3 PRUEBAS DE MEMORIA VS TIEMPO

En esta sección encontramos una gráfica de la memoria en función de tiempo para cada uno de los 3 algoritmos utilizados, las pruebas se realizaron bajo la librería Mprof que por medio de unas etiquetas en el código, llamando a los métodos de cada algoritmo y por consola, ejecuta los 3 algoritmos de manera secuencial con pocos intervalos entre ellos con un pequeño sleep para dar precisión a los resultados.



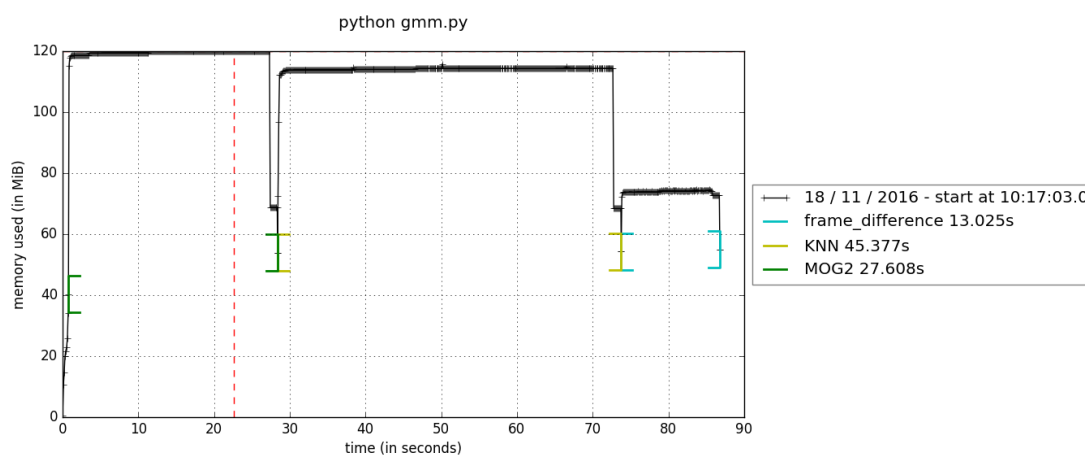


Figura 6-3 Memoria vs tiempo clip 1

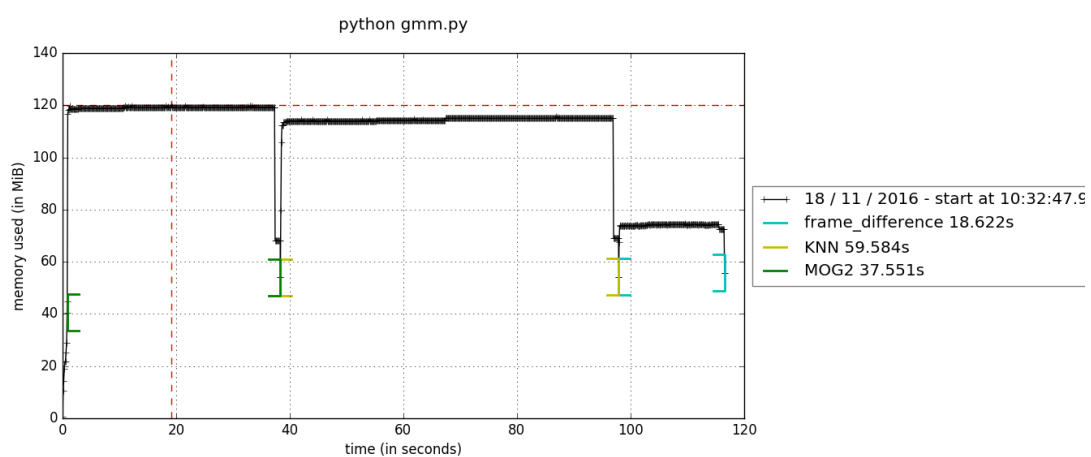


Figura 6-4 Memoria vs tiempo clip 2

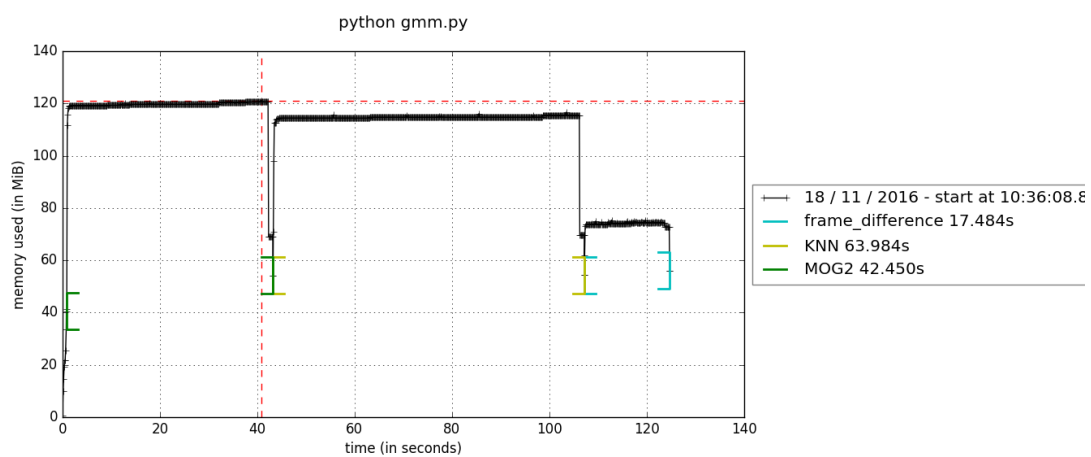


Figura 6-5 Memoria vs tiempo clip 3

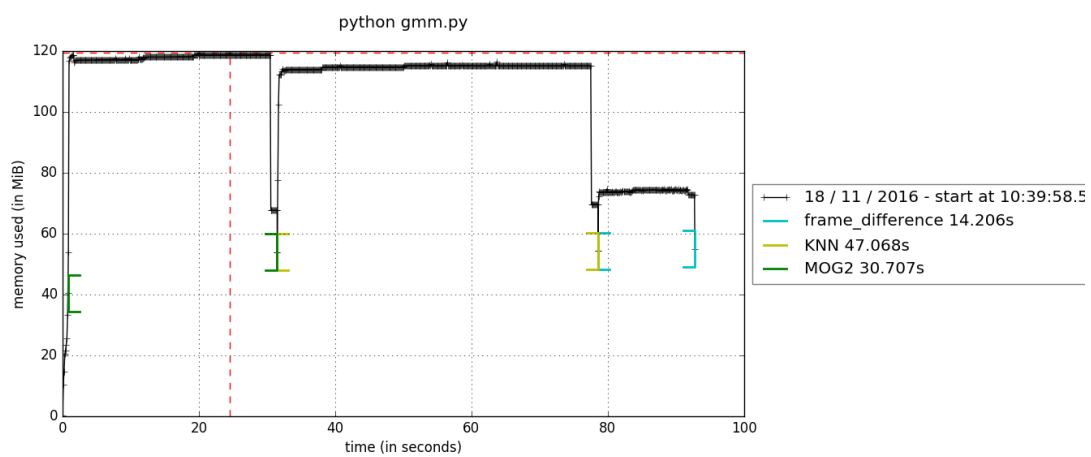


Figura 6-6 Memoria vs tiempo clip 4



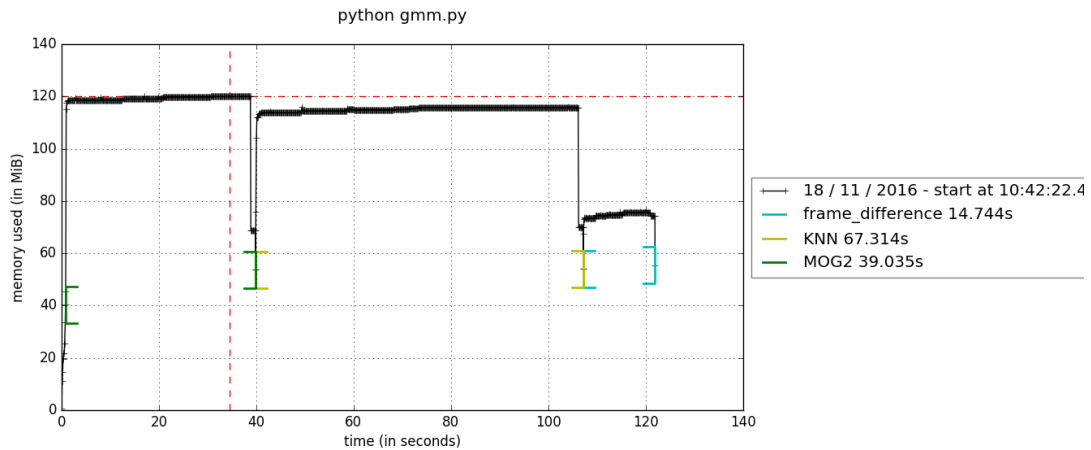


Figura 6-7 Memoria vs tiempo clip 5

### Primer conjunto de videos.

Un primer vistazo a los tiempos de los videos nos indican que bajo condiciones de poca iluminación, el algoritmo de frame difference tiene un promedio de tiempo de 15,616 segundos superando por un amplio margen a los otros algoritmos, en la parte de memoria el algoritmo de frame difference también toma la delantera con un consumo de 75MB de memoria y referente a los otros 2 algoritmos el MOG2 tiene un menor tiempo de ejecución pero el KNN tiene un menor consumo de memoria.

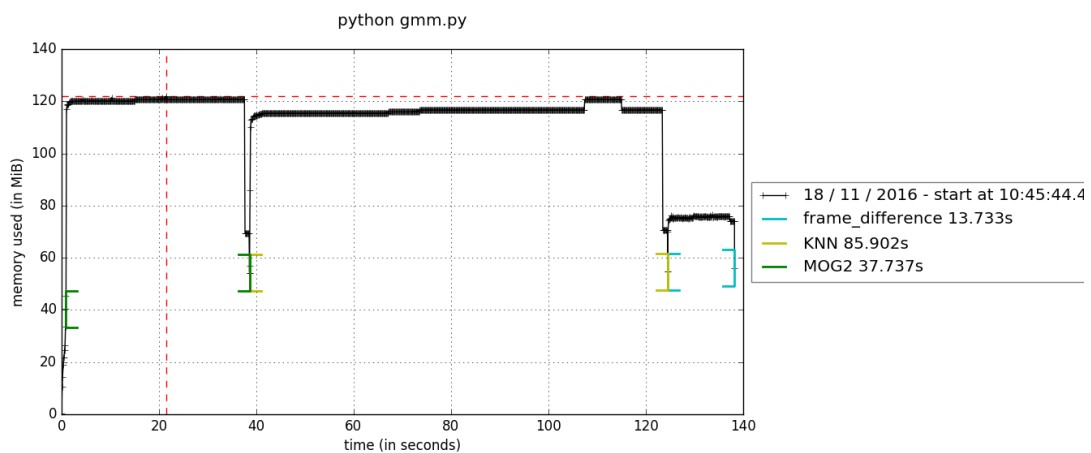


Figura 6-8 Memoria vs tiempo clip 6

### Segundo conjunto de videos.

En este caso, así como en el anterior, el algoritmo de frame difference se lleva los mejores valores con un tiempo de 13.733 segundos y un consumo de memoria de 78 concluyendo en este caso que con un video concurrido el algoritmo MOG2 tiene al igual que en los anteriores un alto coste de memoria y el de KNN tiene un cambio brusco en consumo de

memoria que luego se ve estabilizado, en esta ocasión al igual que antes el MOG tuvo mejor tiempo que KNN pero mayor consumo de memoria, en este caso las diferencias en memoria se redujeron y en el máximo mencionado alcanzan a ser iguales en un valor de 119MB.

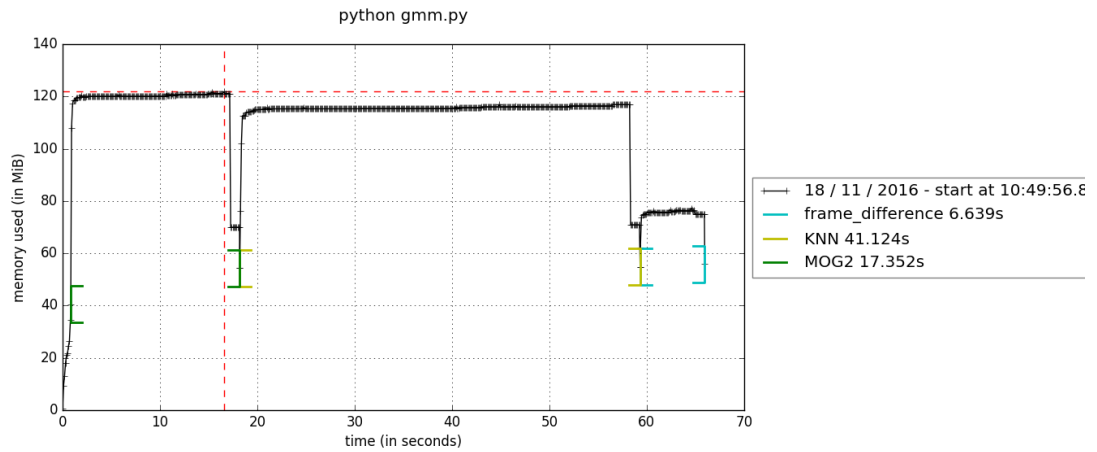


Figura 6-9 Memoria vs tiempo clip 7

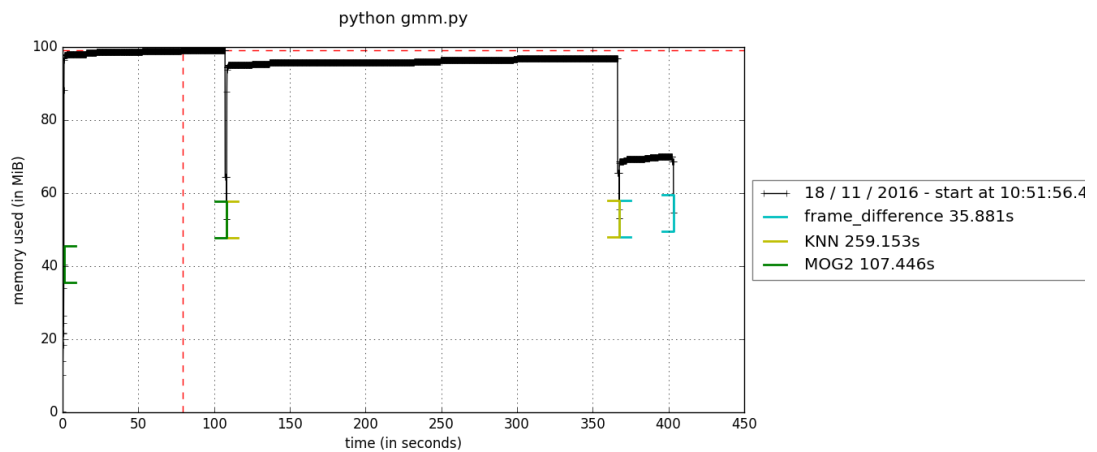


Figura 6-10 Memoria vs tiempo clip 8

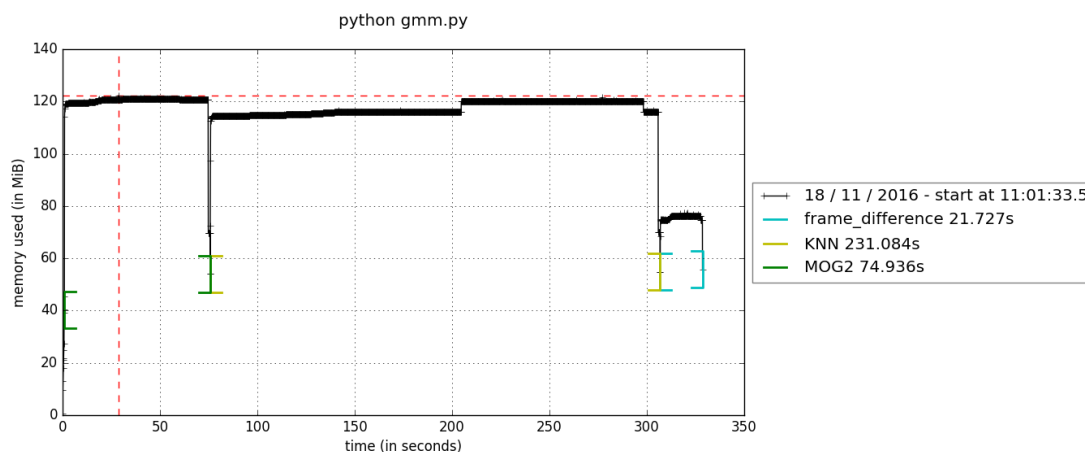


Figura 6-11 Memoria vs tiempo clip 9

### Tercer conjunto de videos

En este caso enfrentamos los algoritmos a los videos con malas condiciones (movimiento del fondo, mala iluminación, ruido, etc.) y que el frame difference mantiene su bajo consumo de memoria y su bajo tiempo de procesamiento, se nota en el primer clip que el KNN tiene un crecimiento acelerado en memoria al inicio y que luego se estabiliza, en el tercer clip hay un salto en consumo de memoria pasando de un valor de 115-116 a un valor de 119-120, el segundo clip fue el que más exigió a los algoritmos y las ventajas del frame difference en este aspecto se ven reflejadas en mayor medida.

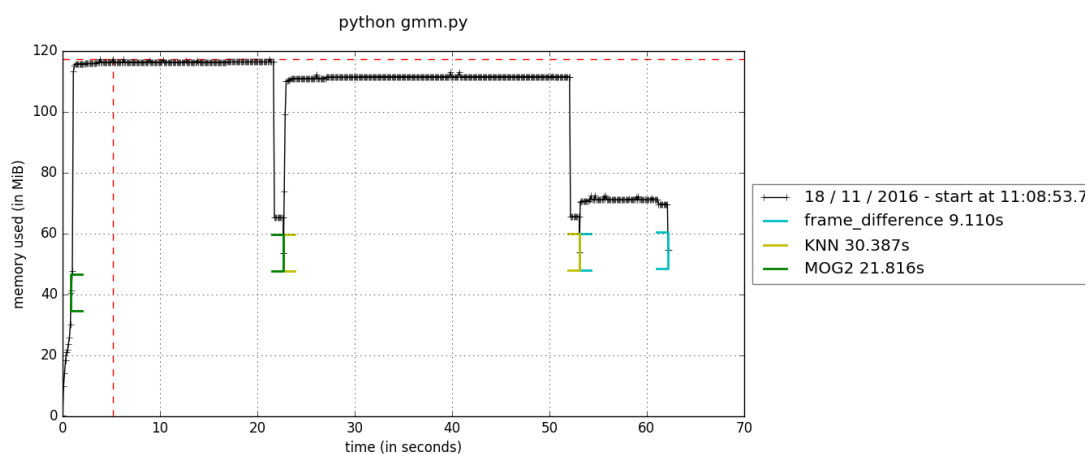


Figura 6-12 Memoria vs tiempo clip 10

### Cuarto conjunto de videos

En este caso el consumo de memoria del algoritmo KNN bajó con referencia a los datos previos, en esta situación al igual que en las pasadas la victoria en cuestión de memoria y tiempo es para el frame difference con 9.110 segundos y un consumo de memoria de 73Mb

en comparación con los otros dos (KNN con 30.387 segundos y 112Mb y MOG2 con 21.816 segundos y 120 Mb)

6.4 PRUEBAS DE PRECISION Y RECALL

Se evaluaron los algoritmos de sustracción de fondo descritos arriba en el mismo conjunto de clips de las pruebas de tiempo vs memoria en los diferentes videos y se sacó un promedio general para simplificar los resultados. Al igual que en las pruebas anteriores los algoritmos se separaron por grupos.



Figura 6-13 MOG sobre el clip 10

Pruebas Conjunto de clips 1-5

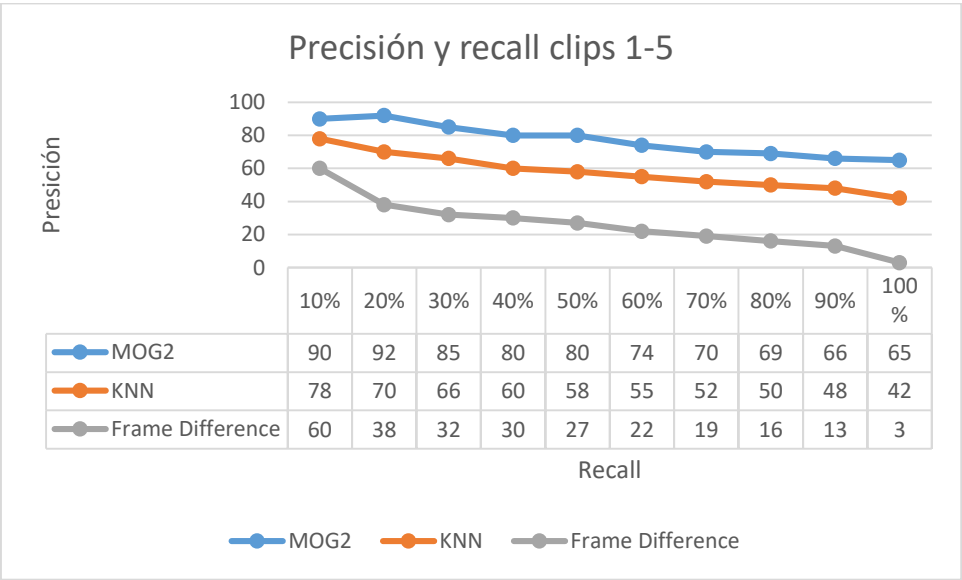


Tabla 6-1 precisión y recall sobre los primeros 5 clips (promedio)

Como se puede observar en la tabla 4-1 en este caso el algoritmo más preciso bajo estas (poca iluminación y bajo flujo poblacional) es el algoritmo MOG2, se observa que en un principio el algoritmo KNN tenía precisión similar pero al avanzar el recall la precisión

empieza a disminuir con una pendiente mayor a la de los otros 2, estando incluso por debajo de frame difference en una ocasión.

Pruebas Conjunto de clip 6

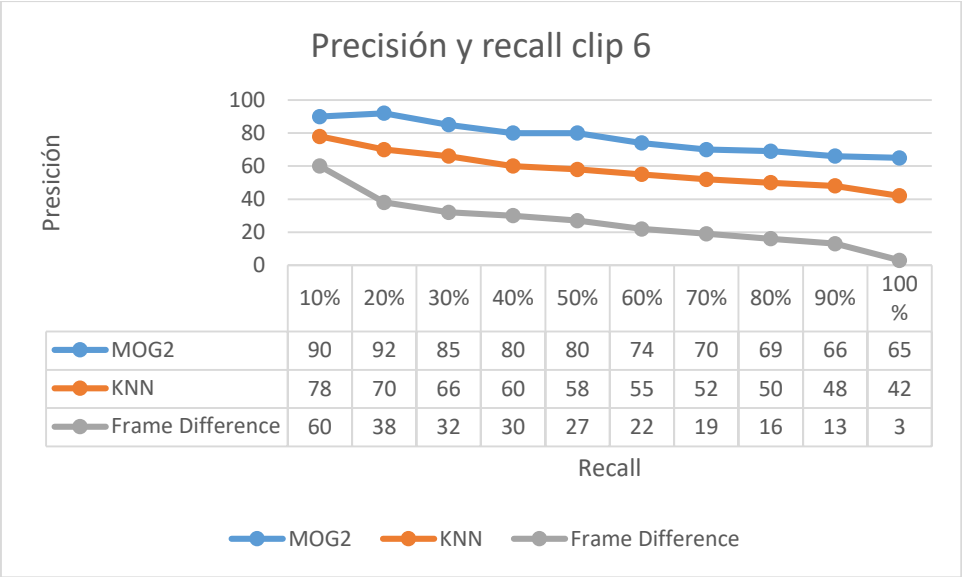


Tabla 6-2. Precisión y recall del 6 clip

En este caso encontramos un video con una gran densidad poblacional en una calle concurrida y con una variación de iluminación poco visible, en esta oportunidad a pesar del frame difference haber sido el victorioso en las pruebas de memoria y tiempo, en esta ocasión fue el peor de los 3 de lejos llegando a una cifra de precisión cercana a 0, se observa también que en este caso el algoritmo MOG2 pierde tiene en unos tramos pendiente positivas y negativas, esto se debe al flujo variable de personas que hay en el clip y solo la tiene este algoritmo pues es quien diferencia de mejor manera las siluetas de personas en las masas, en la figura 4-12 se observa la ejecución del algoritmo frame difference.



Figura 6-14 Algoritmo Frame difference en el clip 6



Figura 6-15 Algoritmo KNN en el clip 6

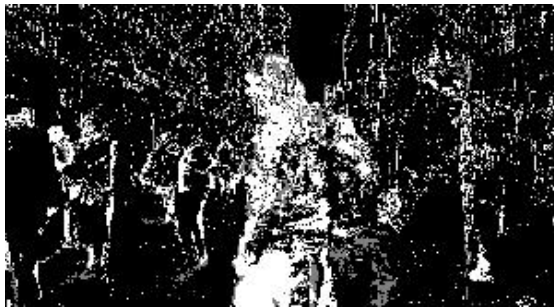


Figura 6-16 Algoritmo MOG2 en el clip 6

### Pruebas Conjunto de clips 7 8 y 9

En este caso con los clips con condiciones desfavorables como se puede observar en las figuras 4-13 4-14 y 4-15 los 3 algoritmos tienen problemas para procesar el fondo pues al ser dependientes de frames anteriores por ser estadísticos el KNN y MOG tienen en algunos frames rastros de fondo de frames anteriores, el que menos problemas adquiere en esta tarea por su simplicidad es el de frame difference pues solo trabaja con 2 frames, en el caso de los otros 2 al tener que hacer estimados en cada frame tiene dificultades a la hora de procesar, estas fueron evidentes en las pruebas de memoria y tiempo, al ser el algoritmo de frame difference el más simple, tiene poca tolerancia a cambios de iluminación pero sorprendentemente se desenvuelve bien con fondos variables al solo depender del frame inmediatamente anterior, aun así por la falta de robustez mencionada el número asciende considerablemente.



Figura 6-17 Algoritmo KNN en el clip 8





Figura 6-18 Algoritmo MOG2 en el clip 9



Figura 6-19 Algoritmo Frame Difference en el clip 7

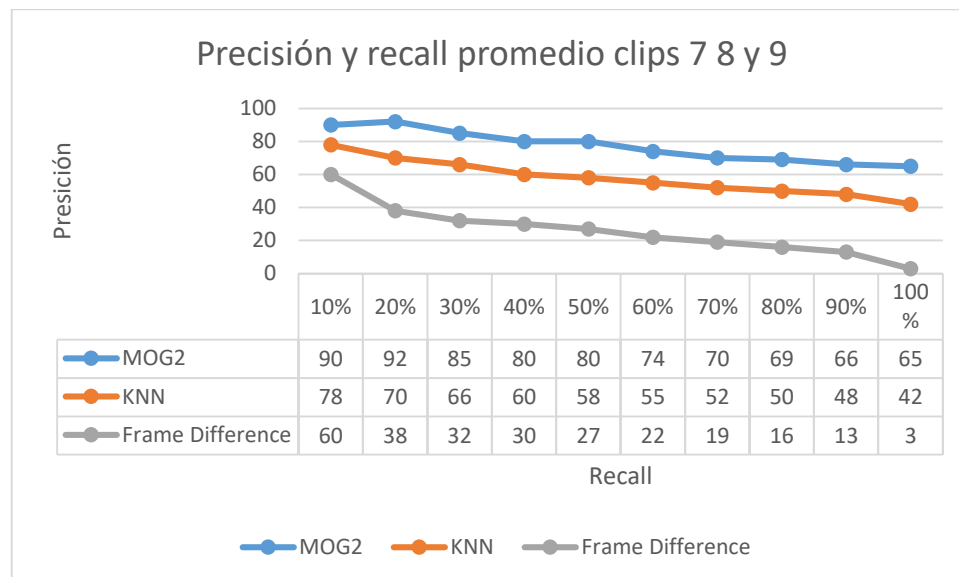


Tabla 6-3 Precisión y recall de clips 7, 8 y 9

## Pruebas Conjunto de clip 10



Figura 6-20 Algoritmo KNN en el clip 10

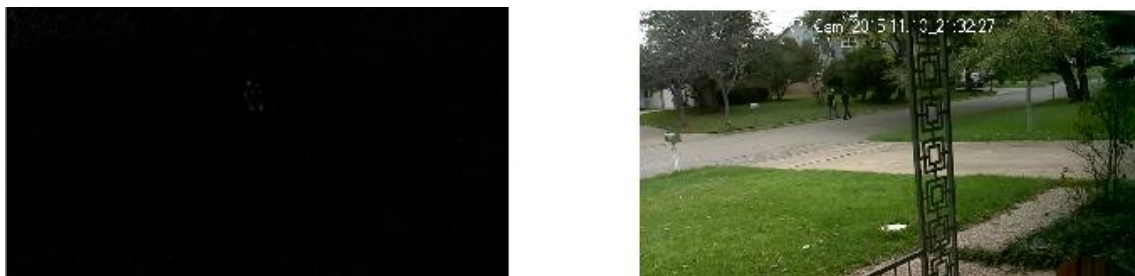


Figura 6-21 Algoritmo Frame Difference en el clip 10

En este caso se observa que el algoritmo de KNN tiene inconvenientes con el ondeo de los árboles, el algoritmo de frame difference por otro lado discrimina los cambios leves y da un binario con muy poca información rescatable, catalogándose como un algoritmo poco recomendado para este tipo de situaciones, por otro lado el MOG de la figura 4-11 presenta la mejor mascara de los 3 algoritmos, en este caso al igual que en los anteriores, como se vio en las pruebas vuelve a coincidir el tiempo de ejecución lento para el KNN y el MOG y un gran procesamiento y poco coste de memoria para el frame difference, llegando en ocasiones a terminar el proceso sin que fuera posible tomar varias muestras en una ejecución, analizando el grafico de precisión y recall en este caso el peor algoritmo es el de Frame difference con una precisión bastante mala descartándose totalmente para conteo en estas condiciones, ganando en este caso el algoritmo de MOG siendo este el recomendado para este tipo de casos también en la literatura.



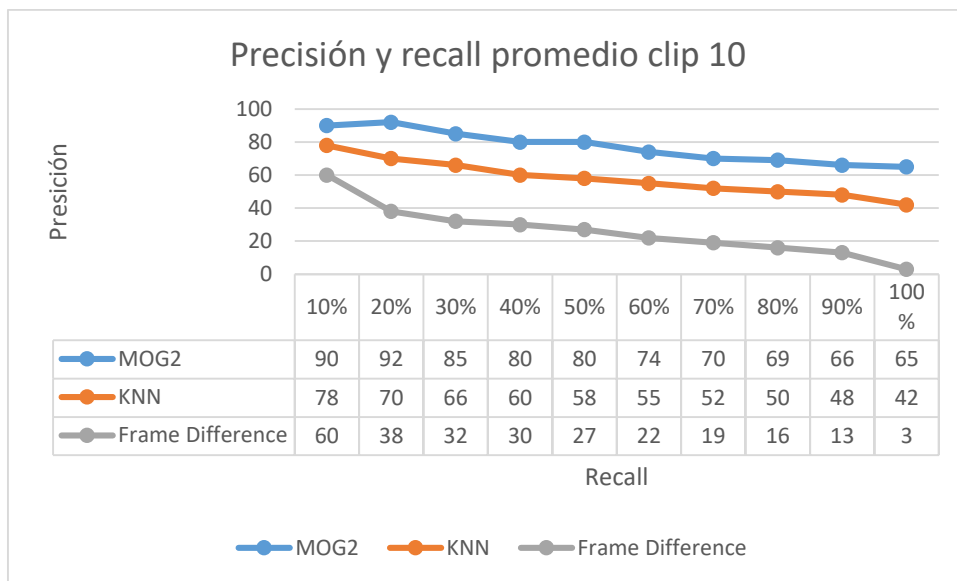


Tabla 6-4 Precisión y recall de clip 10

# CONCLUSIONES

En este proyecto se diseñaron e implementaron 3 algoritmos de sustracción de fondo y se dará un análisis de si son o no aptos para el conteo de personas, los criterios de evaluación fueron tiempo, memoria y precisión y recall.

El algoritmo de MOG fue el que tuvo un mejor desempeño de los 3, tiene un tiempo de procesamiento aceptable, pero su coste de memoria fue el superior, lo que hace a este algoritmo el vencedor es sus buenos resultados en las pruebas de precisión, mostrando que es un algoritmo que se adapta a muchas de los desafíos encontrados en ambientes reales, por lo tanto el algoritmo recomendado para conteo de personas que satisface la mayor cantidad de escenarios es el de MOG, se recomienda que si alguno de los ambientes donde se implementara no está tomado como adecuado en la literatura, se busquen soluciones alternativas en conjunto con otros algoritmos como los híbridos explicados en el capítulo 3.

El algoritmo Frame Difference fue el mejor en términos de consumo de memoria y de tiempo de procesamiento, superando a los otros 2 en la totalidad de los videos, pero opacando estos resultados con su mal desempeño en las pruebas de precisión, obviando detalles de información importante en clips como en el 10 en donde la percepción de movimiento en este caso es casi nula por lo tanto el binario no es apto en ese tipo de ambientes para conteo de personas. El algoritmo bajo condiciones buenas de iluminación y con un fondo poco cambiante no tiene problemas y por lo tanto es recomendado usarse en casos donde el dispositivo no tenga muchos recursos. Otro detalle a resaltar es que el buen desempeño de este algoritmo se da con una distancia no muy grande entre las personas y la cámara como las del clip número 6.

El último algoritmo el de KNN fue de desempeño medio en todas las pruebas, tiene un buen desempeño cuando no hay muchos píxeles en el frente, es decir, un flujo no muy grande de personas, no es recomendable usarse para hacer sustracción de fondo para conteo de personas pues es opacado en todos los aspectos por los otros 2 algoritmos.

Este trabajo permite resolver problemas tanto de seguridad como de distribución del espacio, pues abre las puertas a la utilización de minería de datos con video sin necesidad de sensores y de fácil acceso para empresas medianas y pequeñas como pueden ser parqueaderos, centros comerciales, supermercados, y en mayor medida, la implementación en el transporte masivo en el cual se envié automáticamente flota de buses en las estaciones que están saturadas, y se manejen mejor los intervalos en estaciones de poca población para así mejorar el rendimiento del transporte en la ciudad.

El trabajo desarrollado sirve de guía para trabajos similares y afines que utilicen procesamiento de video y sustracción de fondo, tanto en la forma de utilizar las técnicas como su implementación en el lenguaje Python.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Escobar & P. Ríos. Caracterización de flujos peatonales en el campus universitario Meléndez. *Universidad del Valle. Facultad de Ingenierías*, 2008
- [2] D. Chacón. Modelo de la movilidad peatonal en la Universidad del Valle sede Meléndez. *Universidad del Valle. Facultad de Ingeniería*, 2011
- [3] C. Benavides & F. Paz. Modelo clásico de planificación de transporte peatonal en la Universidad del Valle sede Meléndez. *Universidad del Valle. Facultad de Ingenierías*, 2012
- [4] L. Rodríguez. Modelo de seguimiento de la movilidad peatonal en la Universidad del Valle sede Meléndez. *Universidad Del Valle*, 2012
- [5] T. Bouwmans & F. Porikli, B. Hoferlin, A. Vacavant. Background modeling and foreground detection for video surveillance, 2015
- [6] T. Shireen, M. Khaled, S. El, H. Sumaya. Moving Object Detection in Spatial Domain using Background Removal Techniques – State-of-Art, 2008.
- [7] M. Harville. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *IVC*, 2004, 22(2):127–142.
- [8] L. Maddalena & A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.
- [9] S. Ching, S. Cheung & C. Kamath. Robust techniques for background subtraction in urban traffic video, 2007.
- [10] R. Ibarra, J. Hoyos & C. Molina. Estrategias de decisión para evaluar la accesibilidad peatonal, 2010.
- [11] B. Xie, V. Ramesh, and T. Boult. Sudden illumination change detection using order Consistency. *Image and Vision Computing*, 22(2):117–125, February 2004.
- [12] M. Singh, V. Parameswaran, and V. Ramesh. Order consistent change detection via fast statistical significance testing. *IEEE Computer Vision and Pattern Recognition Conference, CVPR 2008*, June 2008.
- [13] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving Video. *Signal Processing*, pages 165–180, 1993.
- [14] T. Aach, A. Kaup, and R. Mester. Change detection in image sequences using Gibbs random fields: a Bayesian approach. *IEEE Workshop Intelligent Signal Processing and Communications Systems*, October 1993.
- [15] T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using Markov Random Fields. *Signal Processing Image Communication*, 7:147–160, 1995.

- [16] R. Mester, T. Aach, and L. Duembgen. Illumination-invariant change detection using a Statistical colinearity criterion. DAGM 2001, pages 170–177, September 2001.
- [17] T. Aach, L. Dumbgen, R. Mester, and D. Toth. Bayesian illumination-invariant motion Detection. IEEE International Conference on Image Processing, ICIP 2001, 3:640–643, October 2001.
- [18] T. Aach, D. Toth, and R. Mester. Motion estimation in varying illumination using a Total least squares distance measure. Picture Coding Symposium, PCS 2003, pages 145–148, April 2003.
- [19] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, C. Rosenberger. Comparative study of background subtraction algorithms, 2010.
- [20] Z. Zivkovic, F. Hejiden. Efficient adaptive density estimation per image pixel for the task of background subtraction. 2004.
- [21] D. Alex, A. Wahi. BSFD: Background subtraction frame difference algorithm for moving obect detection and extraction, 2014.
- [22] P. Kaew, R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection, 2001.
- [23] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction, 2004.
- [24] T, Bouwmans. Non Parametric GPU accelerated background modeling of complex scenes.
- [25] T, Bouwmans. GPU implementation for background-foreground-separation via robust PCA and Robust Subspace Tracking.
- [26] D. Barragán, C. Sandoval. Identificación y registro de matrículas vehiculares independiente del color mediante técnicas de visión artificial y reconocimiento de patrones, 2005.
- [27] C. Wren and A. Azarbayejani. Pfinder : Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780–785, July 1997.
- [28] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure. Robust foreground extraction technique using Gaussian family model and multiple thresholds. Asian Conference on Computer Vision, ACCV 2007, pages 758–768, November 2007.
- [29] H. Lin, T. Liu, and J. Chuang. A probabilistic SVM approach for background scene initialization. International Conference on Image Processing, ICIP 2002, 3:893–896, September 2002.
- [30] J. Wang, G. Bebis, and R. Miller. Robust video-based surveillance by integrating target detection with tracking. IEEE Workshop on Object Tracking and Classification

Beyond the Visible Spectrum in conjunction with CVPR 2006, June 2006.

[31] A. Tavakkoli, M. Nicolescu, and G. Bebis. Novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. International Symposium on Visual Computing, ISVC 2006, pages 40–49, November 2006.

[32] D. Culbrik, O. Marques, D. Socek, H. Kalva, and B. Furht. A neural network approach to Bayesian background modeling for video object segmentation. International Conference on Computer Vision Theory and Applications, VISAPP 2006, February 2006.

[33] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. International Conference on Vision Systems, ICVS 1999, January 1999.

[34] T. Bouwmans. Subspace learning for background modeling: A survey. Recent Patents on Computer Science, 2(3):223–234, November 2009.

[35] D. Culbrik, O. Marques, D. Socek, H. Kalva, and B. Furht. Neural network approach to background modeling for video object segmentation.

[36] R. Luque, D. Lopez-Rodriguez, E. Merida-Casermeyro, and E. Palomo. Video object segmentation with multivalued neural networks. IEEE International Conference on Hybrid Intelligent Systems, HIS 2008, pages 613–618, 2008.

[37] R. Luque, E. Dominguez, E. Palomo, and J. Munoz. A neural network approach for video object segmentation in traffic surveillance. International Conference on Image Analysis and Recognition, ICIAR 2008, pages 151–158, 2008.

[38] D. Mukherjee and J. Wu. Real-time video segmentation using Student’s  $t$  mixture model. International Conference on Ambient Systems, Networks and Technologies, ANT 2012, pages 153–160, 2012.

[39] L. Guo and M. Du. Student’s  $t$ -distribution mixture background model for efficient object detection. IEEE International Conference on Signal Processing, Communication and Computing, ICSPCC 2012, pages 410–414, August 2012.

[40] Y. He, D. Wang, and M. Zhu. Background subtraction based on nonparametric Bayesian estimation. International Conference Digital Image Processing, July 2011.

[41] T. Haines and T. Xiang. Background subtraction with Dirichlet processes. European Conference on Computer Vision, ECCV 2012, October 2012.

[42] J. Ding, M. Li, K. Huang, and T. Tan. Modeling complex scenes for accurate moving objects segmentation. Asian Conference on Computer Vision, ACCV 2010, pages 82–94, 2010.

[43] Z. Liu, W. Chen, K. Huang, and T. Tan. Probabilistic framework based on KDE-GMM hybrid model for moving object segmentation in dynamic scenes. International Workshop on Visual Surveillance, ECCV 2008, October 2008.

- [44] Z. Liu, K. Huang, and T. Tan. Foreground object detection using top-down information based on EM framework. *IEEE Transactions on Image Processing*, 21(9):4204–4217, September 2012.
- [45] O. Barnich and M. Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences. *International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, pages 945–948, April 2009.
- [46] T. Bouwmans. Background subtraction for visual surveillance: A fuzzy approach. *Handbook on Soft Computing for Video Surveillance*, Taylor and Francis Group, 5, March 2012.
- [47] L. Maddalena and A. Petrosino. Multivalued background/foreground separation for moving object detection. *International Workshop on Fuzzy Logic and Applications, WILF 2009*, 5571:263–270, June 2009.
- [48] F. El Baf, T. Bouwmans, and B. Vachon. A fuzzy approach for background subtraction. *International Conference on Image Processing, ICIP 2008*, pages 2648–2651, October 2008.
- [49] H. Kashani, S. Seyedin, and H. Yazdi. A novel approach in video scene background estimation. *International Journal of Computer Theory and Engineering*, 2(2):274–282, April 2010.
- [50] E. Fauske, L. Eliassen, and R. Bakken. A comparison of learning based background subtraction techniques implemented in CUDA. *NAIS 2009*, pages 181–192, 2009.
- [51] M. Gong and L. Cheng. Real time foreground segmentation on GPUs using local online learning and global graph cut optimization. *International Conference on Pattern Recognition, ICPR 2008*, December 2008.
- [52] L. Cheng and M. Gong. Real time background subtraction from dynamics scenes. *International Conference on Computer Vision, ICCV 2009*, September 2009.
- [53] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli. Real-time discriminative background subtraction. *IEEE Transaction on Image Processing*, 20(5):1401–1414, December 2011.